1982

Finite element program for calculating flows in turbomachines with results for NASA Task-1 compressor.

Ferguson, Julian A.

Monterey, California. Naval Postgraduate School

2.79

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

FINITE ELEMENT PROGRAM FOR CALCULATING FLOWS IN TURBOMACHINES WITH RESULTS FOR NASA TASK-1 COMPRESSOR

BY

Julian A. Ferguson III

October 1982

Thesis Advisor: Raymond P. Shreeve

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Finite Element Program for Calculating Flows in Turbomachines with Results for NASA Task-1 Compressor	5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis October 1982 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Julian A. Ferguson III	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE October 1982 13. NUMBER OF PAGES 247
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Element Method

Flow in Turbomachines

Compressor Flow Field

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A general mesh generation code (MESHGEN) and finite element flow solver (TURBO) for calculating the flow development through axial turbomachines are fully documented. The finite element approach followed Hirsch and Warzee. Excellent results were obtained for the NASA Task-1 compressor operating with subsonic flow conditions. Construction of the code will allow

straightforward extension to transonic flows, turbine stages and multiple stage machines.

Approved for public release; distribution unlimited

Finite Element Program for Calculating Flows in Turbomachines with Results for NASA Task-1 Compressor

by

Julian A. Ferguson III
Lieutenant, United States Navy
B.S., Auburn University, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

and

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
October 1982

ABSTRACT

A general mesh generation code (MESHGEN) and finite element flow solver (TURBO) for calculating the flow development through axial turbomachines are fully documented. The finite element approach followed Hirsch and Warzee. Excellent results were obtained for the NASA Task-1 compressor operating with subsonic flow conditions. Construction of the code will allow straightforward extension to transonic flows, turbine stages and multiple stage machines.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Description	Units
ψ	Stream function	lbm/sec
α	Absolute flow angle whose tangent is the ratio of the absolute tangential-to-meridional velocity	degrees
β	Relative flow angle whose tangent is the ratio of the absolute tangential-to-meridional velocity	degrees
δ	Deviation angle, difference in flow angle and camber-line angle at trailing edge in cascade projection tangential-to-meridional velocity	degrees
κ_1	Angle between tangent to the blade meanline and the axial direction	degrees
ϕ	Camber angle, difference between angles in cascade projection of tangents to camber line at extremes of camber-line arc	degrees
σ	Solidity, ratio of chord to blade spacing	dimensionless
$\tilde{\omega}$	Total-pressure-loss coefficient	dimensionless
T	Temperature	°R
P	Pressure	psia
ρ	Density	lbm/ft^3

Subscripts

t	Total conditions
1	blade leading edge
2	blade trailing edge
E	Over an element

Superscripts

e	For an element

I. INTRODUCTION

A. STATEMENT OF TASK

The original task of this research project was to continue the work of Macchi [Ref. 1] and Cirone [Ref. 2] in the development of a turbine prediction computer code for the Turbopropulsion Laboratory at the Naval Postgraduate School. An analysis of the referenced code by Ferguson [Ref. 3] indicated that a significant amount of work remained to be done in order to make the program operational. In the author's opinion the task could be better accomplished through the use of a different solution technique. After additional study and review of similar work [Refs. 4, 5, 6 and 7] it was decided that a finite element approach to the problem would be adopted. A program developed by Gavito [Ref. 8], which followed the work of Hirsch and Warzee [Ref. 4], was selected as the basis for development of the computer code described in the sections that follow. However, Gavito's program was formulated as a compressor performance prediction which, as it was reported, had not given results similar to those obtained by Hirsch and Warzee. Thus the first goal of the project became the development of an axial compressor prediction code that could produce results comparable to those published by Hirsch and Warzee. The second goal was to revise and document the program so that its

application to any compressor and its extension to turbine analysis could be carried out without excessive difficulty.

B. DESCRIPTION OF THE PROBLEM

One purpose of conducting a through-flow analysis is to predict the performance of a turbomachine under design and off-design operating conditions. Through the combination of a mathematical model and empirically determined correlations, it is possible to provide the engineer with a tool that will determine the effects of variations in design parameters and analyze the performance of an existing machine.

The first problem addressed in the formulation of a performance prediction code is that of expressing the analysis in a form that can be accurately and efficiently solved by computer methods. Most methods for through-flow calculations are based on the classic work of Wu [Ref. 9] which stated that the equations of fluid flow in turbomachines can be solved on the intersecting sets of stream surfaces known as the S1 family and S2 family of stream surfaces (Fig. 1). In general the intersection of a S1 surface and a S2 surface is a twisting line with three dimensional variations. However, if an axisymmetric assumption is made, the S2 surface will lie on a meridional plane and the directional derivatives on the S2 surface become the $\partial(\ )/\partial r$ and $\partial(\ )/\partial z$ in cylindrical coordinates. As shown by Smith [Ref. 10], circumferentially averaged equations with an axisymmetric

flow assumption can be used to a good first approximation for the through-flow analysis.

Three general methods of solving the so-called radial equilibrium equation of flows in turbomachines which results from the axisymmetric assumption can be found in the literature. The first is called the streamline curvature method [Refs. 1, 2 and 11]. The method derived its name from the fact that the radius of curvature of the streamline is an integral part of the formulation. The second, a matrix method, applies a finite differences technique to the radial equilibrium equation, normally after it has been reduced to a Poisson form [Refs. 12 and 13]. The third method is the finite element method which was used in the present work.

In the mid-1970's, Hirsch and Warzee [Ref. 4] first reported the application of the finite element method to solution of the axisymmetric radial equilibrium equation. They applied the finite element technique to solve the equation expressed in quasi-harmonic form in terms of the stress function. They published extensive comparisons of the predictions obtained using their method with measurements obtained on several machines under various operating conditions. In general, the method produced excellent results for compressors and turbines of single and multi-stage configurations. It was this demonstrated ability of the method over

such a wide range of parameters that let to its selection for use in the present project.

In the sections which follow, the development of programs MESHGEN and TURBO, which are based on the work of Hirsch and Warzee [Ref. 4], is documented. Comparisons are given of the results obtained when the program was applied to analyze the NASA Task-1 compressor with results obtained by the referenced authors.

II. MATHEMATICAL MODEL

The equation of motion for a fluid has the general form (Vavra [Ref. 14])

$$(\partial\vec{V}/\partial t) + (\vec{V}\cdot\nabla)\vec{V} = -\nabla p/\rho + \vec{f}_f - \nabla(gz) \tag{1}$$

Using the vector identity

$$(\vec{V}\cdot\nabla)\vec{V} = \nabla(V^2/2) - (\vec{V} \times \nabla \times \vec{V}) \tag{2}$$

Eq. (1) can be written as

$$\partial\vec{V}/\partial t + \nabla(V^2/2 + gz) = -\nabla p/\rho + \vec{f}_f + (\vec{V} \times \nabla \times \vec{V}) \tag{3}$$

The first law of thermodynamics for a fluid particle can be written as

$$Tds = dh - dp/\rho \tag{4}$$

which, along an elemental path length $d\vec{r}$ in a fluid field implies that

$$T(d\vec{r}\cdot\nabla s) = (d\vec{r}\cdot\nabla)h - (d\vec{r}\cdot\nabla)p/\rho \tag{4a}$$

or

$$d\vec{r} \cdot (\nabla h - T\nabla s - \nabla p/\rho) = 0 \tag{5}$$

Since dr is not equal to zero in general, in a homogeneous fluid flow the first law may be expressed as

$$\nabla h - T\nabla s - \nabla p/\rho = 0 \tag{6}$$

Substituting Eq. (6) into Eq. (3) yields

$$\partial\vec{V}/\partial t + \nabla(h + V^2/2 + gz) = T\nabla s + \vec{f}_f + \vec{V}\times\nabla\times\vec{V} \tag{7}$$

For steady, inviscid flow Eq. (7) reduces to

$$\nabla H = T\nabla s + (\vec{V}\times\nabla\times\vec{V}) \tag{8}$$

As shown by Hirsch and Warzee [Ref. 15], Eq. (8) can be revised to describe the flow through blade rows by introducing a circumferential averaging process and assuming that the flow is axisymmetric at the averaged condition. The averaged equation can be expressed as

$$-(\vec{V}\times\nabla\times\vec{V}) = T\nabla s - \nabla H + \vec{F}_b + \vec{F}_d \tag{9}$$

where F_b is the body force representing the action of the blades on the flow and F_d represents the dissipative force whose work generates the irreversible entropies. The F_d forces are considered to be uniformly distributed in the tangential direction and proportional to the loss coefficients. Equation (9) leads to the following three equations in cylindrical coordinates (with $\partial(\ )/\partial\theta = 0$)

$$(V_u/r)\left(\partial(rV_u)/\partial r\right) - V_z\left((\partial V_r/\partial z) - (\partial V_z/\partial r)\right) =$$

$$\partial H/\partial r - T(\partial s/\partial r) - F_{b,r} - F_{d,r} \tag{10a}$$

$$(V_z/r)\left(\partial(rV_u)/\partial z\right) + (V_r/r)\left(\partial(rV_u)/\partial r\right) = F_u \tag{10b}$$

$$V_r\left((\partial V_r/\partial z) - (\partial V_z/\partial r)\right) - (V_u/r)\left(\partial(rV_u)/\partial z\right) =$$

$$\partial H/\partial z - T(\partial s/\partial z) - F_z \tag{10c}$$

Equation (10a) expresses the radial equilibrium of the meridional through-flow and it is the governing equation to be solved for the velocity components. Equations (10b) and (10c) determine the tangential and axial components of the forces once Eq. (10a) has been solved.

For the solution of Eq. (10a) to have physical meaning, care must be taken to ensure that continuity is satisfied throughout the field. In general, the continuity equation can be expressed as

$$\partial\rho/\partial t + \nabla\cdot(\rho\vec{V}) = 0 \tag{11}$$

which for steady, circumferentially averaged flow can be written as

$$(\partial/\partial r)(\rho r b V_r) + (\partial/\partial z)(\rho r b V_z) = 0 \tag{12}$$

where b is a blockage factor describing the reduction in the flow area in the tangential direction due to the presence of rotor and stator blades. The tangential blockage is approximated by

$$b = 1 - t/s \tag{12a}$$

where t is the blade thickness and s is the blade spacing. As will be discussed in the description of subroutine INPUT, this factor will be modified to account for the end-wall boundary layer contractions. A stream function, ψ, can be introduced and defined so that Eq. (12) is automatically satisfied as follows:

$$\partial\psi/\partial r = \rho r b V_z \tag{13a}$$

$$\partial\psi/\partial z = -\rho r b V_r \tag{13b}$$

After the substitution of Eqs. (13a) and (13b) into Eq. (10a) one is assured of the implicit satisfaction of continuity as the radial equilibrium equation is solved explicitly. Equation (10a) may now be written as

$$(\partial/\partial r)\Big((1/\rho r b)(\partial\psi/\partial r)\Big) + (\partial/\partial z)\Big((1/\rho r b)(\partial\psi/\partial z)\Big) =$$
$$(1/V_z)\Big((\partial H/\partial r) - T(\partial s/\partial r) +$$
$$(V_u/r)\big(\partial(rV_u)/\partial r\big)\Big) - F_{b,r} - F_{d,r} \tag{14}$$

The equation is written in a slightly different form for solution in rotor regions where rothalpy remains constant along a streamline. The definition of rothalpy, H_R, given by

$$H_R = H - UV_u \tag{15}$$

is substituted into Eq. (14) and the terms in brackets on the right-hand side become

$$\Big[\partial H_R/\partial r - T(\partial s/\partial r) - (W_u/r)\big(\partial(rV_u)/\partial r\big) - F_{b,r} - F_{d,r}\Big] \tag{16}$$

The significance of the $F_{b,r}$ and $F_{d,r}$ terms can be analyzed in the following manner. The body force F_b acts in a direction normal to the mean blade surface, which for radial blading is in the circumferential direction. The term $F_{b,r}$ accounts for the body force component in the radial direction that results when blade lean is present. For most blading this is a small term that may be neglected.

Similarly, $F_{d,r}$ is the contribution of the dissipative forces in the radial direction for non-cylindrical stream surfaces. This contribution can normally be neglected for axial-flow machines, which is the case treated here. (Note that the two body force terms are included in the analysis for machines in which the magnitudes of these forces are significant.) With these simplifications the radial equilibrium equation may be written in the form

$$(\partial/\partial r)\Big((1/\rho rb)(\partial\psi/\partial r)\Big) + (\partial/\partial z)\Big((1/\rho rb)(\partial\psi/\partial z)\Big) =$$
$$(1/V_z)\Big((\partial H/\partial r) - T(\partial s/\partial r)\Big)+(V_u/r)\Big(\partial(rV_u)/\partial r\Big) \qquad (17)$$

with the appropriate modifications for solution in a rotor region.

III. FINITE ELEMENT METHOD

A. INTRODUCTION

The finite element method is a numerical procedure that is particularly well suited to solving problems in continuum mechanics, which invariably involve equations expressed in differential form. As stated by Cook [Ref. 16], the essence of the finite element method is the "piecewise approximation of a function ϕ, by means of polynomials, each defined over a small region (element) and expressed in terms of nodal values of the function."

In order to understand the finite element method and the solution techniques employed in the computer program reported herein, one must first have a complete understanding of the basic element, the terminology used to describe the element, and the relationship between the element and the remainder of the solution domain. The complete problem is solved in a piecewise manner, in which the solution of the derived governing relationship over the discrete region of an element is sought to determine the contribution of the element to the overall solution. Figure 2 illustrates the single element as it is used in the present work and the nomenclature for the element on what is referred to as the "local domain". The number scheme to employ is arbitrary, limited only by the requirement that the system remain

consistent from element to element. Figure 3 shows a vertical stack of three elements to show how elements are connected in what is known as a "global domain". Table 1 lists the relationship between the two reference systems, known as the connectivity. The connectivity is important because the solution of a problem over a computational region involves a careful summation of the local contributions of each element to the global equations. The summation process is tracked by the connectivity. Again the global numbering scheme is arbitrary, influenced mainly by considerations of computer storage and computational efficiency.

The key concept to be grasped is that the finite element method is a series of local solutions that are coupled together through the connectivity relationships to form a solution for the entire computational domain. A more detailed description of the finite element method is contained in Refs. 16 through 18.

B. APPLICATION OF THE WEIGHTED RESIDUAL PROCESS

A standard weighted residuals process was used to transform Eq. (17) into a form that can be solved by numerical techniques. As a first step, the equation was written in a more compact form as

$$(\partial/\partial r)[k(\partial\psi/\partial r)] + (\partial/\partial z)[k(\partial\psi/\partial z)] + f = 0 \tag{18}$$

where

$$k(r,z) = (1/\rho rb) \tag{19}$$

$$f(r,z) = (1/V_z)[T(\partial s/\partial r) - \partial H/\partial r + (V_u/r)(\partial(rV_u)/\partial r)] \qquad (20)$$

with the boundary conditions

$$k(\partial\psi/\partial n) + \alpha_1(\psi - \psi_0) = 0 \qquad (21)$$

on the associated exterior surface S. Equations (18) and (21) may be rewritten as

$$(1/r)\{(\partial/\partial r)[k(\partial\psi/\partial r)] + (\partial/\partial z)[k(\partial\psi/\partial z)] + f\} = 0 \qquad (22)$$

in the volume, V, and

$$(1/r)[k(\partial\psi/\partial n) + \alpha_1(\psi - \psi_0)] = 0 \qquad (23)$$

on the surface, S. An approximation, $\tilde{\psi}(r,z)$, of the unknown solution is searched for such that the corresponding weighted residual, $\overline{R}$, is equal to zero. Analytically this is expressed as

$$\overline{R} = \int_V W(r,z)\ R_V(r,z)\ dV + \int_S W(r,z)\ R_S(r,z)\ dS = 0 \qquad (24)$$

where W(r,z) is the (known) weight function and R_V and R_S are the so-called "residuals" in the volume and on the surface, respectively. As the sum of R_V and R_S approaches zero, the approximation, $\tilde{\psi}$, approaches the exact solution, ψ, with R_V and R_S defined to be identically zero if $\tilde{\psi} = \psi$. By defining R_V to be equal to the left-hand side of Eq. (22) and R_S to be equal to the left-hand side of Eq. (23), Eq. (24) can be written as

$$\int_\Omega -W(r,z)[(\partial/\partial r)\{k(\partial\psi/\partial r)\} + (\partial/\partial z)\{k(\partial\psi/\partial z)\} + f]2\pi d\Omega$$
$$+ \int_C Wk(\partial\psi/\partial n)2\pi dC = 0 \qquad (25)$$

Integration of the first term of Eq. (25) by parts yields

$$\int_{\Omega} [k\{(\partial\psi/\partial r)(\partial W/\partial r) + (\partial\psi/\partial z)(\partial W/\partial z)\} - Wf]\, d\Omega = 0 \qquad (26)$$

if ψ is selected to equal ψ_0 along the corresponding part of the boundary. The second term of Eq. (26) reduces to zero through the proper application of the boundary conditions.

The boundary conditions must be satisfied in different ways for different portions of the boundary. Along the inlet where $\alpha_1 = 0$ the conditions may be satisfied by specifying $(\partial\psi/\partial n)$ to be zero or by specifying the nodal values of ψ if the inlet conditions are conducive to calculating ψ for each node. Along the shroud and along the hub the value of ψ must be specified as $\psi = (m/2\pi)$ at the shroud and $\psi = 0$ at the hub. For nodes at the exit plane, the condition that $(\partial\psi/\partial n) = 0$ is required.

C. APPLICATION OF THE FINITE ELEMENT METHOD

The first step is to discretize the region into subregions or elements. Within each element the unknown stream function, ψ, and the coordinates r and z are assumed to have the following polynomial variations:

$$\psi(r,z) = \sum_{i}^{n} \psi_i N_i(\xi,\eta) \qquad (27a)$$

$$r = \sum_{i}^{n} r_i N_i(\xi,\eta) \qquad (27b)$$

$$z = \sum_{i}^{n} z_i N_i(\xi,\eta) \qquad (27c)$$

where n = number of nodes in the element

N_i = the shape or (trial) function for node i

ψ_i = value of ψ at node i

r_i = value of r at node i

z_i = value of z at node i

Equations (27b) and (27c) imply a geometrical as well as functional transformation or mapping, as shown for the present case in Fig. 4.

The second step in the process is the selection of the weight and shape functions. The shape functions are defined when the particular type of finite element is selected for the computational grid [Ref. 16]. The eight-noded quadrilateral was used in the present program and its associated shape functions were entered in a subroutine. The weight function is independent of the shape function and may be chosen at the discretion of the individual. In the present case the standard Galerkin technique was employed and therefore, the weight functions were defined as being equal to the shape functions, so that

$$W(r,z) = N(r,z) \tag{28}$$

Equation (26) may now be expressed in the following form:

$$\iint_E \left\{ k \left[(\partial N_j/\partial r) \sum_i^n (\partial N_i/\partial r) + (\partial N_j/\partial z) \sum_i^n (\partial N_j/\partial z) \sum_i^n (\partial N_i/\partial z) \right] - N_j f(r,z) \right\} d\Omega = 0 \tag{29}$$

where $\int_E$ represents the integral over an element. In matrix notation this becomes

$$[K]^e\{\delta\}^e = \{f\}^e \tag{30}$$

where

$$k^e_{ij} = \int_E k(r,z)[(\partial N_j/\partial r)(\partial N_i/\partial r)+(\partial N_j/\partial z)(\partial N_i/\partial z)]d\Omega \tag{31a}$$

$$f^e_i = \int_E N_i \; f(r,z) \; d\Omega \tag{31b}$$

and

$$\delta_i = \psi_i \tag{31c}$$

The summation of the elemental contributions over the entire region yields the global system of equations needed to solve the problem. In matrix notation the global system of equations is expressed as

$$[K]\{\delta\} = \{f\} \tag{32}$$

where

$$[K] = \sum_i^m [K]^e_i \tag{33a}$$

$$\{\delta\} = \sum_i^m \{\delta\}^e_i \tag{33b}$$

$$\{f\} = \sum_i^m \{f\}^e_i \tag{33c}$$

and m = number of elements in the mesh

$\delta_i = \psi_i$

$[K]$ = system's stiffness matrix

$\{f\}$ = system's right-hand side vector

Since k_{ij} and f_i depend on the unknown solution ψ, Eq. (32) is a nonlinear differential equation to be solved by an iterative procedure. The details of the procedure are contained in section V.

D. NUMERICAL INTEGRATION TECHNIQUE

In general, problems are analyzed using a coordinate system in which the boundary conditions can be written and satisfied in the simplest possible way. For problems with irregularly shaped boundaries and/or mixed conditions along different portions of the boundary, obtaining numerical solutions in the original coordinate system can be a formidable task. Very often a scheme must be found to transform the derived equations into a new coordinate system that conforms to the requirements of standard numerical techniques. Traditional transformation techniques tend to be complicated exercises in algebra that require extensive reformulation for each geometry or type of boundary condition. The power of the finite element method is the automatic inclusion of a transformation of the geometry and the function to a rectangular domain where a variety of integration techniques may be employed. This can be seen in Fig. 4, which illustrates what is implied by Eqs. (27a), (27b), and (27c). The method can handle extremely complicated boundary conditions

and shapes with ease and is limited only by the type of element selected by the individual.

The quadratic properties of the eight-node element allows curved boundaries in the physical domain so long as the section of the boundary included within a single element does not have a point of inflection. The use of a quadratic element also ensures continuity of the approximated function along the elemental boundaries regardless of the direction of approach from within the mesh. The specific numerical technique used in the program is discussed in the following section.

1. Stiffness Matrix Evaluation

In section C the following relationship was derived for the individual elements of the stiffness matrix, [K]:

$$k^e_{ij} = \int_E k(r,z)[(\partial N_j/\partial r)(\partial N_i/\partial r) + (\partial N_j/\partial z)(\partial N_i/\partial z)]d\Omega \qquad (31a)$$

In order to take advantage of well established numerical integration techniques, Eq. (31a) must be transformed from the (z,r) domain and its irregular elemental boundaries to the rectangular (ξ,η) domain. Equations (27a) through (27c) describe the variation of the function and the (r,z) coordinate values in the (ξ,η) plane. In order to transform Eq. (31a) it is necessary to determine the relationship of the variations of the derivatives in the two domains. These relationships can be derived in a straightforward manner through the use of the chain rule as follows:

$$(\partial N_i/\partial\xi) = (\partial N_i/\partial z)(\partial z/\partial\xi) + (\partial N_i/\partial r)(\partial r/\partial\xi) \tag{34}$$

and

$$(\partial N_i/\partial\eta) = (\partial N_i/\partial z)(\partial z/\partial\eta) + (\partial N_i/\partial r)(\partial r/\partial\eta) \tag{35}$$

Equations (34) and (35) can be combined in matrix form as

$$\begin{Bmatrix} \dfrac{\partial N}{\partial\xi} \\ \\ \dfrac{\partial N}{\partial\eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial z}{\partial\xi} & \dfrac{\partial r}{\partial\xi} \\ \\ \dfrac{\partial z}{\partial\eta} & \dfrac{\partial r}{\partial\eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial N_i}{\partial z} \\ \\ \dfrac{\partial N_i}{\partial r} \end{Bmatrix} \tag{36}$$

Through the selection of the type element to be used in the mesh, $N_i(\xi,\eta)$ is a known function [Ref. 16], which makes possible the computation of the left-hand side vector for any point within the element boundaries. Similarly, by taking the appropriate derivatives of Eqs. (27b) and (27c) the 2x2 matrix, known as the Jacobian matrix [J], can be determined. It follows that the r and z derivatives of the shape/weight functions can be determined for any point of an element from

$$\begin{Bmatrix} \dfrac{\partial N_i}{\partial z} \\ \\ \dfrac{\partial N_i}{\partial r} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial z}{\partial\xi} & \dfrac{\partial r}{\partial\xi} \\ \\ \dfrac{\partial z}{\partial\eta} & \dfrac{\partial r}{\partial\eta} \end{bmatrix}^{-1} \begin{Bmatrix} \dfrac{\partial N_i}{\partial\xi} \\ \\ \dfrac{\partial N_i}{\partial\eta} \end{Bmatrix} \tag{37}$$

An examination of Eqs. (27a) through (27c) shows that once the derivatives of the shape/weight functions are known for a point then it is a simple procedure to determine the

derivatives of any other elemental property that has a value specified at the nodes.

The final relationship that is needed for the transformation is the relationship between the differential change in the coordinate directions of the (r,z) plane and the (ξ,η) plane. As shown by Kaplan [Ref. 19], the required relationship is

$$dzdr = |J|d\xi d\eta \tag{38}$$

Equation (31a) can now be transformed for integration in the (ξ,η) plane to the form

$$k_{ij}^{e} = \int_{-1}^{1}\int_{-1}^{1} \sum_{k}^{n} N_k k_k \left([B]^T [B] \right) |J| d\xi d\eta \tag{39}$$

where

$$\left[B\right] = \left[J\right]^{-1} \begin{Bmatrix} \dfrac{\partial N}{\partial \xi} \\ \\ \dfrac{\partial N}{\partial \eta} \end{Bmatrix} \tag{40}$$

and

$$k = (1/\rho rb) \tag{41}$$

The Gauss-Legendre method of numerical integration was used to obtain a solution to Eq. (39). It was selected because its determined accuracy was easily determined and the simple summing procedure used in the solution could be efficiently coded. A one dimensional example is used here to illustrate the use of the method.

The definite integral

$$I = \int_{-1}^{1} \phi(\xi) d\xi \tag{42}$$

may be written in the form

$$I \simeq W_1\phi_1 + W_2\phi_2 + W_3\phi_3 + \ . \ . \ . + W_n\phi_n \tag{43}$$

where $\phi_i = \phi(\xi_i)$

W_i = Gaussian weight fuction for ξ_i

The values of the points, called abscissas, and their corresponding weighting function values are catalogued for use. The number of points to be used is determined by the order of the function to be approximated. In general, a polynomial of (2n - 1) is integrated exactly by the use of n Gauss points. In two dimensions the Gauss method can be written as

$$I = \int_{-1}^{1} \int_{-1}^{1} \phi(\xi, \eta) d\xi d\eta \tag{44}$$

which can be written as

$$I \simeq \sum_{i}^{n} \sum_{j}^{m} W_i W_j \phi(\xi_i, \eta_j) \tag{45}$$

Equation (39) can now be written in a form that can be coded for solution by the computer as

$$k_{ij}^{e} = \sum_{k} \sum_{\ell} W_k W_\ell \sum_{m} N_m k_m ([B]^T [B]) |J| \tag{46}$$

The scheme used in the program is a two dimensional, three-point Gauss integration. A detailed description of the evaluation of $[B]^T[B]$ and the method used to obtain k_m is contained in the description of subroutine STIFF.

2. Right-hand Side Vector Evaluation

In Section C the following relationship for f was derived:

$$f_i^e = \int_E N_i \; f(r,z) \; d\Omega \tag{31b}$$

By using the techniques of Section D1, Eq. (31b) can be replaced by

$$f_i^e = \sum_j \sum_k W_j W_k \sum_i N_i \sum_\ell N_\ell F_\ell |J| \tag{47}$$

A more detailed description of the specific methods used to solve Eq. (47) is contained in the description of subroutine FCAL.

IV. DESCRIPTION OF PROGRAM MESHGEN

A. MAIN PROGRAM DESCRIPTION

MESHGEN was developed in order that the required inputs for the program TURBO could be generated in a fast, accurate, and conceptually correct manner. The program generates an eight-node isoparametric element mesh, computes the related connectivity matrix, defines the type of region enclosed by each element, computes the tangential blockage factor and an estimate of the stream function for each node in the mesh, and computes the thermodynamic conditions and velocity at the inlet. The inputs required to operate the program are the mass flow rate, total temperature, and total pressure at the inlet, the blading and machine geometries, RPM, Cp, γ, R, and scaling constants for the plot of the mesh. The blading geometries must be coded in the program as a subroutine that uses approximations or design information to express the blade variables as functions of radius. The user provides the other information in response to interactive prompts. The program has two modes of operation, one which generates a complete mesh and all of the associated information and another which uses a previously generated mesh to compute the changes in specific arrays that result from a change in the inlet conditions.

The program is completely general and may be used for either single-stage axial compressor or turbine, and, with very minor modifications, can be expanded for use with multi-stage machines. The mesh size that can be generated by the program is limited only by computer storage considerations. To use the program for another machine, the user is required to replace subroutine TASK1 with a subroutine that can compute the tangential blockage factor, b, for the desired blading. The functioning of the program and its subroutines for both modes of operation is described in the section B.

The program's algorithm in outline form is as follows:

Algorithm:

Determine the value of the appropriate operating conditions and whether a new mesh is desired (Subroutine INIT1).

Obtain the coordinates of the super element corners and a description of the division of the super elements into the final mesh. Compute the storage allocation parameters (Subroutine INPUT).

Compute the (r,z) coordinates for all nodes in the mesh (Subroutine TMESH).

Compute the connectivity relationships for the mesh and determine the beginning and ending node numbers for the rotor and stator (Subroutine CONEC).

Compute the array of node numbers where the value of ψ is to be specified. Compute an initial estimate of the nodal stream function distribution and call subroutine FLOFCT to determine the inlet conditions (Subroutine INIT2).

Compute the nodal tangential blockage factor, b, for the rotor and stator nodes (Subroutine TASK1).

Place the computed values in disk storage (Subroutine FILGEN).

Plot the generated mesh on the Tektronix 618 terminal for inspection (Subroutine MPLOT).

END

B. SUBROUTINE DESCRIPTIONS

1. Subroutine INIT1

Subroutine INIT1 obtains the value of thermodynamic variables and plotting parameters that are required for the program in either mode of operation. The user is required to provide the values of the mass flow (lbm/sec), total temperature (°R), and total pressure (psia), ratio of specific heats (γ), gas constant (ft-lbf/lbm-°R), and the specific heat at constant pressure (BTU/lbm-°R). The scaling constants are convenient values of r and z used to frame the plot of the mesh. If a new mesh is not desired, the program exits the subroutine.

Subroutine INIT1 determines if a new mesh is to be generated by the response of the user to an interactive question. If a mesh is to be generated, the subroutine obtains some preliminary information about the flow region. Figure 5 shows how the user must first divide the flow region into a coarse mesh known as super elements, recording the (z,r) coordinates of the corner points. The minimum number of super elements for a single-stage compressor is five so that the three duct regions, the rotor, and the stator may be represented. The maximum number of super elements and the subsequent division into mesh elements is limited only by the

storage limitations of the machine. In practice, the maximum number of elements is limited by the number of equations that may be solved by the program TURBO. It is also limited by the fact that only one super element may be used to describe a rotor or duct region and that the super elements for these regions may only be further subdivided into a single column of elements. The latter restrictions are for compatibility with the computation procedures used in the program TURBO. A decision must then be reached on what subdivision of the super elements will provide a mesh that is sufficiently fine to yield accurate results efficiently. Once the flow region has been divided into super elements and a determination as to the total number of rows and columns of mesh elements has been made, the user can input the appropriate values in response to the prompts provided by the program.

2. Subroutine INPUT

Subroutine INPUT uses interactive prompts to obtain a description of the turbomachine's flow passage geometry and the desired mesh characteristics. The user must provide the program with the coordinate values of the super element nodes as shown in Fig. 5. The values are entered as nodal pairs on a station-by-station basis. The first (z,r) coordinate entered is the node on the shroud and the second lies on the hub. The program then asks the user to identify the type of region enclosed in each super element and into how many columns each super element is to be divided.

Enough information is now available for the program to compute and store the information required for the program TURBO. Subroutine INPUT stores the responses to the interactive prompts, determines the values of the storage location pointers, and determines if any storage limitation has been exceeded. If any storage limitation is exceeded, the subroutine calls the appropriate error subroutine to halt execution. The interactive portion of the subroutine is omitted if a new mesh is not desired. However, the values of the pointers are calculated and storage requirements are evaluated as before. A listing of the pointers and the corresponding variables is given in Appendix A.

3. <u>Subroutine TMESH</u>

Subroutine TMESH computes the nodal coordinate values from the information obtained in subroutines INIT1 and INPUT. The subroutine uses linear interpolation in the axial direction and a quadratic scheme in the radial direction. The radial interpolation scheme maintains the difference in the squares of the radius of the nodes as a constant. This allows the assumption of equal mass flow between the nodes for uniform axial velocity which is used to determine the initial estimate of the nodal stream function distribution. The nodes are numbered with the assumption that the fluid flow is from left-to-right in the mesh and that the number of columns of elements is greater than or equal to the number of rows of elements. The node at the inlet shroud is labeled

1 and the node at the outlet hub is labeled n for an n node mesh. The numbering proceeds on a column-by-column basis from top-to-bottom. The total number of rows, columns, and mesh elements is displayed to the user. The mesh computations are omitted if a new mesh is not created and elemental count information is displayed as before.

4. Subroutine CONEC

Subroutine CONEC generates the connectivity matrix for the mesh. The connectivity matrix is used to keep track of which nodes are assigned to which elements and the arrangement of the assigned nodes within the element. The connectivity relationships for a three element stack is shown in Fig. 3. Additionally, the subroutine determines the first and last nodes of the rotor and stator.

5. Subroutine INIT2

Subroutine INIT2 stores the node numbers for nodes where the value of ψ is specified as a known quantity. The array is used in the program TURBO to apply the boundary conditions. The array is not computed if a new mesh is not desired. For either mode of operation, subroutine INIT2 computes the values of the inlet thermodynamic variables, the inlet axial velocity, and an initial estimate of the nodal stream function distribution. Subroutine FLOFCT is used to calculate the inlet conditions and is described in the next section. The initial stream function is computed from the boundary conditions at the shroud and hub. Along

the shroud, ψ is specified to be equal to $(\dot{m}/2\pi)$ and along the hub to be zero. The value of ψ along the inlet is determined by a linear interpolation because of the quadratic node spacing in the radial direction. The remaining nodal values of ψ are obtained by an assumption that ψ is a constant along the streamwise boundaries of the elements. The assumption is obviously in error, but it observes the boundary conditions and provides a reasonable first estimate to begin the iteration scheme used in the program TURBO.

6. Subroutine FLOFCT

Subroutine FLOFCT computes the inlet conditions for a passage with a specified geometry, mass flow rate, total temperature, total pressure, and an assumed uniform inlet velocity. The method followed is the "total flow function" formulation proposed by Shreeve [Ref. 20]. The total flow function is defined as the ratio of the mass flux to the limiting or stagnation mass flux. The following definitions and equations are required for the method:

$$V_t = [2H]^{0.5} \tag{48}$$

$$X = V/V_t \tag{49}$$

$$T/T_t = 1 - X^2 \tag{50}$$

$$p/p_t = (1 - X^2)^{\frac{\gamma}{(\gamma-1)}} \tag{51}$$

$$\rho/\rho_t = (1 - X^2)^{\frac{1}{(\gamma-1)}} \tag{52}$$

From the definition of the total flow function, ϕ, it follows that

$$\phi = \rho V/\rho_t V_t = X(1 - X^2)^{\frac{1}{\gamma-1}} \tag{53}$$

The value of ϕ at the inlet can be calculated at the inlet from the assumed uniform conditions by the expression

$$\phi_1 = \dot{m}/(\rho_t V_t A) \tag{54}$$

The value of X at the inlet is found through the following Newtonian iteration:

$$\phi_1 = \dot{m}/(\rho_t V_t A) \tag{54}$$

Assume X = 0.1 to assure the selection of the subsonic root.

Calculate: $$\phi = X(1 - X^2)^{\frac{1}{\gamma-1}} \tag{53}$$

and $$d\phi/dX = \{1/X - 2X/[(\gamma - 1)(1 - X^2)]\}$$

Test $$|\phi_1 - \phi| < \varepsilon$$

If the test fails then calculate

$$X = X + (\phi_1 - \phi)(d\phi/dX)$$

and recalculate ϕ until convergence is reached. Once convergence is reached the inlet conditions are computed by equations (50) through (52).

7. Subroutine TASK1

Subroutine TASK1 computes the nodal tangential blockage factor for the blading of the NASA TASK1 transonic compressor. The value of the blockage factor is determined by

$$b = 1 - t/s \tag{12a}$$

The values of t and s are obtained by approximations to the known blading geometry that are expressed as functions of radius. The maximum thickness of the blade is artificially defined to be at the mid-point of the chordline to ensure that the factor is accounted for in the calculations in the program TURBO. This artificiality could easily be removed through a modification to the axial interpolation scheme used in rotor and stator super elements. Subroutine TASK1 is the only machine dependent subroutine in use in the program and would need to be replaced with an appropriate substitute in order for the program to be used on another machine. The use of the subroutine is omitted if the user does not desire a new mesh.

8. Subroutine FILGEN

Subroutine FILGEN places the computed mesh parameters on disk storage for use in the program TURBO. If the limited mode of operation was selected by the user, the subroutine only updates the values of the parameters that change for a new inlet condition. A listing of the output parameters and their corresponding storage location is given in Appendix B.

9. Subroutine MPLOT

Subroutine MPLOT provides an on-line plot of the computed mesh on the Tektonix 618 graphics terminal. Figure 6 shows the 63 element, 222 node mesh used in the computations of the test cases. The subroutine displayed the mesh through

direct calls to the subroutines of the library plotting package, GRAFF. The subroutine's algorithm is as follows:

Algorithm:

Form two Real*4 arrays from the information in the r coordinate and the z coordinate arrays for plotting compatibility.

Sort the arrays and plot the streamwise boundaries of the mesh elements.

Sort the arrays and plot the transverse boundaries of the mesh elements.

END

10. Subroutine ERR1

Subroutine ERR1 is called by subroutine INPUT if the storage limitation for Real*8 variables has been exceeded. The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The user response would be to increase the value of LIMR if possible or reduce the size of the mesh.

11. Subroutine ERR2

Subroutine ERR2 is called by subroutine INPUT if the storage limitation for Real*4 variables has been exceeded. The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The user response would be to increase the value of LIM4 if possible or reduce the size of the mesh.

12. Subroutine ERR3

Subroutine ERR3 is called by subroutine INPUT if the storage limitation for Integer*4 variables has been exceeded.

The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The user response would be to increase the value of LIMI if possible or reduce the size of the mesh.

V. DESCRIPTION OF PROGRAM TURBO

Program TURBO solves the quasi-harmonic stream function radial equilibrium equation for flow in an axial compressor. The program uses the information computed by the program MESHGEN to calculate the desired thermodynamic information at all nodal points and displays selected values on a graphics terminal for inspection.

A. MAIN PROGRAM DESCRIPTION

The program obtains a solution of the equation

$$[K]\{\psi\} = \{f\} \tag{32}$$

An iterative scheme was adopted for this nonlinear problem, whereby an estimate of the stream function distribution is used to calculate values of the velocity components and thermodynamic variables at the nodes of the mesh. These computed values are then substituted into Eq. (32) and a new value of the stream function distribution is calculated. The estimate of the distribution is compared to the calculated value to determine if the solution has reached convergence according to the following criterion:

$$\varepsilon > \left| \frac{\psi_i^{\,n} - \tilde{\psi}_i^{\,n+1}}{\tilde{\psi}_i^{\,n+1}} \right| \tag{47}$$

where $\psi_i^{\,n}$ = estimate of ψ at node i

$\tilde{\psi}_i^{\,n+1}$ = solution for ψ at node i

If the maximum difference for all nodes is less than ε, the procedure is terminated. If the maximum difference exceeds some specified value of ε, the new estimate of ψ to be used for the next iteration is determined using

$$\psi_i^{\,n+1} = \psi_i^{\,n} + \alpha\left[\tilde{\psi}_i^{\,n+1} - \psi_i^{\,n}\right] \tag{48}$$

where $\psi_i^{\,n+1}$ = new estimate of ψ for the next iteration

α = under relaxation factor required for convergence because of the strong nonlinear properties of Eq. (32).

The process of constructing the inputs required for Eq. (32) is repeated until convergence is obtained. The details of calculating the inputs and constructing the stiffness matrix and the right-hand side vector are contained in descriptions of the program's subroutines.

The program's algorithm follows in outline form:

Obtain the computational constants (SUBROUTINE RDATA).

Determine the values of the pointers used to partition the storage arrays (SUBROUTINE INIT1).

Set the initial values for all storage locations to 0.0 or 0 as appropriate (SUBROUTINE ZEROI).

Recall from storage the externally computed input values and initialize the inlet conditions (SUBROUTINE INPUT).

Calculate a velocity and thermodynamic variable distribution based on the assumed stream function distribution and the inlet conditions (SUBROUTINE DIST).

From the distributions obtained in DIST, calculate the right-hand side vector {f} (SUBROUTINE FCAL).

Using the density and blockage factor distributions, form the stiffness matrix [K] (SUBROUTINE STIFF).

Solve the system of linear equations to obtain a new stream function distribution (SUBROUTINE DSIMQ).

Place the solution vector in its proper storage location (SUBROUTINE REPLA).

Compare the original stream function distribution to the solution vector to determine the maximum difference in the distributions for all nodes (SUBROUTINE TEST).

Determine if the convergence criterion has been satisfied.

If convergence has not been obtained, perform the relaxation iteration to update the estimate of the stream function distribution (SUBROUTINE RELAX), prepare for another cycle (SUBROUTINE NOCON), and then return to SUBROUTINE DIST for further calculations.

If the convergence criterion has been satisfied, print the results (SUBROUTINE OUTPUT) and display selected information on the graphics terminal (SUBROUTINE MPLOT).

END

B. SUBROUTINE DESCRIPTIONS

Sections 1 through 23 provide a detailed description of the subroutines of program TURBO.

1. Subroutine RDATA

Subroutine RDATA is used to store the following computational constants:

(a) Logical Input/Output variable NREAD and NWRITE.

(b) Relaxation factor.

(c) Limits for the storage arrays.

(d) Three-point Gaussian abscissas and weighting values.

(e) Constants used for conversions between different units.

2. Subroutine INIT1

Subroutine INIT1 determines the values of the pointers used to partition the Real*8, Real*4, and Integer*4 arrays and determines whether the storage limitations for any of the arrays has been exceeded. If the storage limitations have been exceeded the subroutine will halt execution by calling the appropriate error subroutine. A listing of the pointers and their corresponding array names is contained in Appendix C.

3. Subroutine ZEROI

Subroutine ZEROI sets the initial value of all arrays equal to 0.0 or 0 as appropriate.

4. Subroutine INPUT

Subroutine INPUT retrieves required input information from its corresponding disk storage location. The information must be placed in storage before running program TURBO. The usual method of generating the information and placing it in storage is through the use of the program MESHGEN.

Subroutine INPUT also initializes the inlet conditions to their proper values, and modifies the nodal blockage factors to account for end-wall boundary layer effects. No attempt was made to include a global method for calculating the blockage factors; rather, a method similar to the one used by Hirsch and Warzee [Ref. 4] was used. The full method

used by Hirsch and Warzee was to artificially reduce the size of the flow passage by reducing the boundaries of the mesh, followed by the application of a general blockage factor to the nodes of the mesh. In the program TURBO, no mesh modifications are made. The procedure followed was to apply a general blockage factor to all nodes, followed by the application of an additional blockage factor to nodes in the outer elements in the rotor, stator, and the passage in between. Though reasonable results were obtained by this method, the handling of the end-wall boundary layers remains the most obvious weakness in the code. This is addressed specifically in section VII.

5. <u>Subroutine DIST</u>

Subroutine DIST calculates the distributions of velocity, density, temperature, pressure, fluid flow angles, entropy, and enthalpy using the known blade and machine geometry, inlet conditions, and the assumed distribution of the stream function. Properties of nodes at the mid-line of the rotor or stator blades were assumed to have a value equal to the average of the inlet and exit conditions of the blade. The elemental calculations are accomplished through the control of subroutines SLINE, DUCT, ROTO, and STAT.

The following is the subroutine DIST's algorithm in outline form:

For each element in the mesh.

Determine type element for appropriate computations.

Duct Elements

For each node at stations 2 and 3:

Determine the location of the streamline and thermodynamic conditions at station one (Subroutine SLINE).

(If the element is along the machine exit plane, ensure that $(\partial\psi/\partial z) = 0$.)

Compute the thermodynamic conditions (Subroutine DUCT).

Assign the appropriate values to the proper storage location.

Rotor Elements

For each node at stations 1:

Determine the location of the streamline at station 3 and the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at stations 1 and 3 (Subroutine SLINE).

Determine the inlet and outlet relative flow angles and the outlet absolute flow angle.

Compute the total-to-total pressure ratio and the adiabatic efficiency for the streamline (Subroutine ROTO).

Assign the appropriate values to the proper storage location.

For each node at stations 2:

Determine the location of the streamline and the thermodynamic conditions and the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at stations 1 (Subroutine SLINE).

Determine the location of the streamline and the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at station 3 (Subroutine SLINE).

Determine the inlet and outlet relative flow angles and the outlet absolute flow angle and the relative deviation angle.

Compute the thermodynamic conditions at station 3 (Subroutine ROTO).

Compute the value of all properties for the node as being the average of the values at station 1 and station 3.

Assign the appropriate values to the proper storage location.

For each node at stations 3:

Determine the location of the streamline and thermodynamic conditions at station 1 and the value of $\partial\psi/\partial z$ and $\partial\psi/\partial r$ at stations 1 and 3 (Subroutine SLINE).

Determine the inlet and outlet relative flow angles and the outlet absolute flow angle and compute the thermodynamic conditions (Subroutine ROTO).

Assign the appropriate values to the proper storage location.

Stator Elements

The stator algorithm is the same as the rotor algorithm except the outlet absolute flow angle is the only angle calculated. The inlet absolute flow angle is determined through interpolation.

6. Subroutine SLINE

In order to understand the functioning of this subroutine and others to follow, one must refer to the nomenclature used to describe the eight-node element. Figures 2 and 3 show the nomenclature clearly and Table 1 demonstrates the connectivity. All of the calculations in the program for the distributions of velocity, flow angles, and thermodynamic properties are founded on the assumption that the points in question lie on the same stream surface. Thus the objective of the subroutine is to obtain the location of a given value of the stream function at a specified station in the flow region. The location of the streamline is required in order to compute the variables used in [K] and {f}.

Through the application of the boundary conditions, the nodes along the shroud are defined to lie on one stream surface and the nodes along the hub are on another. It is possible for all other nodes in the mesh to be on different stream surfaces. For these nodes an interpolation scheme must be followed to find the (ξ,η) coordinates of a specified stream surface at a given station in the mesh.

The solution sequence that the program follows starts at the top element of the first column of elements in the mesh and solves the thermodynamic and velocity conditions for all nodes in the element using the assumed stream function distribution and the specified inlet conditions. When the calculations for the first element are complete, the program continues down the column until the calculations are complete for the element along the hub. The program then sequences to the top element for the next column and continues until the calculations are complete for the last element in the mesh. In this sequence it is always possible to calculate the conditions at station 1 of an element for any interim nodal values of the stream function.

The process will be described by way of an example for one node as shown in Fig. 3.

Example: Find the (ξ,η) coordinates of the streamline that passes through node 7 of element 1, Node (1,7).

From the connectivity relationships it is known that

(Node (1,7)) = (14)

The value of (14) is known and the search is begun to find two nodal vaules of ξ at station 1 that bracket the desired value, (14). The program first tests to see of (Node (1,7)) is greater than (Node (1,5)). In this case it is not and the program would automatically shift and test to see of (Node (1,7)) is greater than (Node (2,5)). In this example the value is larger and the same test would be applied to (node (2,4)). Again the answer would be true. The program would then test to see if the value of (Node (2,3)) were larger than (Node (1,7)). The answer being true would signal the program that the location of the streamline had been bracketed and a half-interval technique would be applied to find the location. As shown in Fig. 4, the value of ξ for all locations along station 1 is -1. This fact is important for two reasons. One, with ξ known the program is only required to iterate on η to obtain convergence. Two, the Kronecker delta property of the shape functions means that only the shape functions at station 1 have nonzero values [Ref. 16]. For the half interval method, the program uses the average η of the most recent bracketing as its estimate for η. In this example the first estimate of η is equal to 0.5 and the solution for ψ at (-1., 0.5) can be written as

$$\psi(-1.,0.5) = N_3(-1.,0.5)\psi_3 + N_4(-1.,0.5)\psi_4 + N_5(-1.,0.5)\psi_5$$

The solution is then compared to ψ(Node (1.7)) to determine if the difference is less than some ε. If the difference exceeds ε, the new estimate for η becomes 0.25 or 0.75 depending on whether the solution is larger than or less than the value of ψ(Node (1,7)). The process is continued until convergence is reached. Once ξ and η for the streamline location at station 1 are known, all of the properties for station 1 can be determined. (The same method is used to find the location of the streamline at station 3 for rotor and stator elements.) Having determined the coordinates of streamlines at all desired locations it is possible to calculate the required $(\partial N_i/\partial r)$ and $(\partial N_i/\partial z)$. The computed inlet and exit coordinates and conditions are then passed to subroutine DIST for use in subroutines DUCT, ROTO, and STAT as appropriate for the calculation of the conditions at Node (1,7).

The following is the subroutine's algorithm in outline form:

Duct Element

If the node being investigated is on station one, exit the subroutine.

For stations two and three, determine the streamline coordinates and the thermodynamic conditions at station one and compute the $\partial\psi/\partial z$ and $\partial\psi/\partial r$ at the node. Exit the subroutine.

Rotor/Stator Element

If the node being investigated is on station one, set all inlet thermodynamic variables equal to the corresponding

nodal value and compute the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at station three. Exit the subroutine.

For station two, determine the streamline coordinates and the thermodynamic conditions at station one and compute the streamline location and the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at station three. Exit the subroutine.

For station three, determine the streamline coordinates and the thermodynamic conditions at station one and compute the $\partial\psi/\partial z$ and the $\partial\psi/\partial r$ at station three. Exit the subroutine.

7. Subroutine DUCT

Subroutine DUCT determines the values of temperature, pressure, and density for the elemental nodes at stations two and three. An iterative procedure is used with the knowledge that angular momentum is a constant in a duct. The initial estimate of the velocity at station 1 is made using the computed values of $\partial\psi/\partial z$, $\partial\psi/\partial r$, r, and b at the node (Subroutine SLINE) and by choosing the estimate of the density at the node to be equal to the density at station 1. The following sequence of calculations is repeated until convergence on a value of the exit velocity:

$$V_{m2} = (1/(\rho_2 r_2 b_2)) * \left[(\partial\psi/\partial z_2)^2 + (\partial\psi/\partial r_2)^2\right]^{0.5}$$

$$\alpha_2 = \tan^{-1}\left[(r_1 V_{m1} \tan\alpha_1)/(r_2 V_{m2})\right]$$

$$T_2 = T_{T1} - (\gamma-1)/2 * \left(V_{m2}^{\,2}(1 + \tan^2\alpha_2)\right)/(\gamma R G_c)$$

$$P_2 = P_{T1}(T_2/T_{T1})**(\gamma/\gamma-1)$$

$$\rho_2 = P_2/(RT_2)$$

$$V_{m2n} = (1/(\rho_2 r_2 b_2)) * \left[(\partial\psi/\partial z)_2^{\,2} + (\partial\psi/\partial r)_2^{\,2}\right]^{0.5}$$

Test if $\left|V_{m2n} - V_{m2}\right| < \varepsilon$

The total conditions are then calculated from the static conditions and the computed velocity.

8. Subroutine ROTO

Subroutine ROTO calculates the change in the relative flow angles, the velocity, and the thermodynamic properties along a streamline across a (compressor) rotor element. The program uses the conditions at station one and the location of the streamline and the partial derivatives of the stream function at station 3, all of which were calculated in subroutine SLINE. The relationships in subroutine ROTO are derived from cascade correlations and known property relationships for a streamline in a rotor.

The first step is the calculation of the inlet and exit relative flow angles. For inlet flow without swirl, the relative inlet flow angle can be calculated using

$$\beta_1 = \tan^{-1}(U_1/V_{m1})$$

The incidence angle is calculated using the known blade geometry and inlet angle, since

$$i = \beta_1 - \kappa_1$$

where κ_1 is the angle formed between the tangent to the blade chordline and the axial direction. In order to calculate the exit relative flow angle one must determine the deviation angle, δ. The program uses the correlations and equations

derived by NASA [Ref. 21]. The specific sequence of equations, using the notation of Ref. 21, is as follows:

$$i_C - i_{2D} = f(M,r)$$

$$i_{2D} = (K_i)_t (K_i)_{SH} (i_0)_{10} + n\phi$$

$$i_{ref} = i_{2D} + \left(i_C - i_{2D}\right)$$

$$\delta_{2D} = (K_\delta)_t (K_\delta)_{SH} (\delta_0)_{10} + (m/\sigma^b) + (i_C - i_{2D})(d\delta/di)_{2D}$$

$$\delta_C - \delta_{2D} = f(M_R, r)$$

$$\delta_{ref} = \delta_{2D} + \left(\delta_C - \delta_{2D}\right)$$

$$\delta = \delta_{ref} + (i - i_{ref})(d\delta/di)_{2D}$$

$$\beta_2 = \beta_1 - \phi - i + \delta$$

The expressions used to approximate the NASA correlation curves were those obtained by Crouse of NASA lewis and were provided to the author by Okiishi [Ref. 22].

When the inlet conditions and relative flow angles are known it is possible to determine the conditions at the rotor element exit. The initial exit velocity is obtained in the same way as in subroutine DUCT. The following sequence of equations is solved iteratively until convergence for the exit velocity is reached:

$$\alpha_2 = \tan^{-1}\left[(\omega r_2 - V_{m2} \tan \beta_2)/V_{m2}\right]$$

$$D = 1 - (W_2/W_1) + (r_1 W_{u1} - r_2 W_{u2})/(2 \quad W_1 \bar{r})$$

$$\tilde{\omega} = \text{curve fit to } \tilde{\omega}(D, \cos \beta, \sigma)$$

$$T_{E1} = T_{R1} + \omega^2(r_2^{\ 2} - r_1^{\ 2})/\text{constant}$$

$$P_{E1} = P_{T1}(T_{E1}/T_{T1})*(\gamma/\gamma-1)$$

$$P_{R1} = P_{T1}(T_{R1}/T_{T1})*(\gamma/\gamma-1)$$

$$P_{E2} = P_{E1} - \tilde{\omega}(P_{R1} - P_1)$$

$$W_2 = V_{m2}/\cos\beta_2$$

$$T_2 = T_{E1} - W_2^{\ 2}/\text{constant}$$

$$P_2 = P_{E2}(T_2/T_{E1})*(\gamma/\gamma-1)$$

$$\rho_2 = P_2/(RT_2)$$

$$V_{min} = \left(1/(\rho_2 r_2 b_2)\right)*\left[(\partial\psi/\partial z)_2^{\ 2} + (\partial\psi/\partial r)_2^{\ 2}\right]^{0.5}$$

$$\text{Test if } \left|V_{min} - V_{m2}\right| < \varepsilon$$

When convergence is reached the total conditions are calculated from the static conditions and the computed velocity. The value of the entropy change is calculated by:

$$S = R\ \ln(P_{T2}/P_{T1}) * \text{constant}$$

9. <u>Subroutine STAT</u>

Subroutine STAT calculates the stator element exit absolute flow angle and thermodynamic conditions using the knowledge that the total enthalpy is a constant across the stator. The initial estimate of the exit velocity is obtained in the same way as in subroutine DUCT. The following sequence of equations is used until convergence for the exit velocity is reached:

$$V_1 = \left[V_{m1}{}^2(1 + \tan^2\alpha_1)\right]^{0.5}$$

$$D = 1 - (V_2/V_1) + (r_1V_{u1} - r_2V_{u2})/(2\,\sigma\,V\,\bar{r})$$

$$\tilde{\omega} = \text{curve fit to } \omega(D,\cos\beta,\sigma)$$

$$P_{T2} = P_{T1} - \tilde{\omega}(P_{T1} - P_1)$$

$$T_2 = T_{T1} - (\gamma-1)/2 * (V_{m2}{}^2(1 + \tan^2 {}_2))/(\gamma\,R\,G)$$

$$P_2 = P_{T2}(T_2/T_{T2})**(\gamma/\gamma-1)$$

$$\rho_2 = P_2/(RT_2)$$

$$V_{min} = (1/(\rho_2r_2b_2))*\left[(\partial\psi/\partial z)_2{}^2 + (\partial\psi/\partial r)_2{}^2\right]^{0.5}$$

$$\text{Test if } \left|V_{min} - V_{m2}\right| < \varepsilon$$

When convergence is reached the total conditions are calculated from the static conditions and the computed velocity. The value of the entropy change is calculated by the same method used by subroutine ROTO.

10. Subroutine FCAL

Subroutine FCAL uses the previously computed distributions of total temperature, entropy, enthalpy, axial velocity, and tangential velocity to compute the right-hand side vector for the global system of equations. In the absolute frame of reference, f(r,z) can be expressed in the form

$$f(r,z) = (1/V_z)\ T(\partial s/\partial r) - \partial H/\partial r + (V_u/r)(\partial(rV_u)/\partial r) \qquad (20)$$

The value of f(r,z) within an element is determined using the following relationships:

$$T(z,r) = \sum_i^n N_i(\xi,\eta)T_i \qquad H(z,r) = \sum_i^n N_i(\xi,\eta)H_i$$

$$s(z,r) = \sum_i^n N_i(\xi,\eta)s_i \qquad V(z,r) = \sum_i^n N_i(\xi,\eta)V_i$$

$$V(z,r) = \sum_i^n N_i(\xi,\eta)V_i \qquad r = \sum_i^n N_i(\xi,\eta)r_i$$

where the value of $N_i(\xi,\eta)$ is determined by the value of ξ and η for a specified Gaussian integration point within the element. The required partial derivatives are found in a simple and direct way. To illustrate, the $(\partial H/\partial r)$ is derived as follows:

$$H(z,r) = \sum_i^n N_i H_i$$

$$(\partial H/\partial r) = \sum_i^n (\partial N_i/\partial r)H_i + \sum_i^n N_i(\partial H_i/\partial r)$$

and since H_i is a constant then

$$\partial H/\partial r = \sum_i^n (\partial N_i/\partial r)H_i$$

where $(\partial N_i/\partial r)$ is found by Eq. (37), and H_i is the appropriate nodal value of H. The radial variations in entropy and angular momentum are found in the same manner. The same procedure is followed for the values of rothalpy and relative tangential velocity for rotor elements.

It is now possible to calculate the quantities in the expression for the right-hand side vector at a point. All that remains is to apply an integration technique to

obtain the value over an element and to assemble the resulting local contributions into the global equations. As shown in section III.D.2, the local contribution for node i in an element can be expressed as:

$$f_i^e = \sum_{jk}\sum W_j W_k \sum_i N_i \sum_\ell N_\ell \; f_\ell \; |J| \tag{47}$$

In the program Eq. (47) is modified to

$$f_i^e = \sum_m^9 W_m \sum_i^n N_i \sum_k^n N_k f_k |J|$$

where the Gaussian abscissas and the corresponding product of the weight functions are grouped into three one-dimensional arrays. At the completion of the summing process, the local contribution for f has been calculated for each node in the element. The global system is then updated by adding the local contributions to the global values through the use of the connectivity relationships.

The following is the subroutine's algorithm in outline form:

Algorithm:

Iterate for each element in the mesh.

Iterate for each Gaussian point.

Find shape functions, $|J|$, and $[J]^{-1}$.

Find V_z, T_T, V_u, r, rV_u, $(\partial s/\partial r)$, and $(\partial H/\partial r)$.

Compute the contributions of the value f at the Gauss point to the value of f at each node of the element.

Upon completion of the Gaussian integration, add the local contribution to the global system.

END

The same algorithm is followed for rotor elements with the appropriate substitutions of H_R and W_u.

11. Subroutine STIFF

Subroutine STIFF uses the computed distributions of density and blockage factors to form the stiffness matrix for the global system of equations. It was shown earlier that the contribution to the elemental stiffness is expressed as

$$k_{ij}^{e} = \int_{E} k(r,z)[(\partial N_j/\partial r)(\partial N_i/\partial r) + (\partial N_j/\partial z)(\partial N_i/\partial z)]d\Omega \qquad (31a)$$

where $$k(r,z) = (1/\rho rb) \qquad (19)$$

Again, the elemental properties are considered to have a polynomial variation of the form

$$\rho(z,r) = \sum_{i}^{n} N_i \rho_i, \quad r_i = \sum_{i}^{n} N_i r_i, \quad \text{and } b(z,r) = \sum_{i}^{n} N_i b_i$$

As shown earlier, Eq. (31a) can be converted by the Gauss-Legendre method to

$$k_{ij}^{e} = \sum_{k}\sum_{\ell} W_k W_\ell \sum_{m} N_m k_m \left([B]^T [B]\right) |J| \qquad (46)$$

The value of k_m is determined from the definitions of ρ, b, and r by the same methods used in subroutine FCAL. The evaluation of $[B]^T[B]$ is a simple matter to perform. The

matrix [B] is simply the column vector $\{(\partial N_i/\partial z), (\partial N_i/\partial r)\}$. Therefore, $[B]^T[B]$ can be written $[(\partial N_i/\partial r)(\partial N_j/\partial r) + (\partial N_i/\partial z)(\partial N_j/\partial z)]$. The value of $(\partial N_i/\partial z)$ and $(\partial N_i/\partial r)$ is found for all nodes in the element in one step through the use of subroutine JACOB. The matrix product can be evaluated at a point (z,r) as

$$\sum_i^n \sum_j^n [(\partial N_i/\partial r)(\partial N_j/\partial r) + (\partial N_i/\partial z)(\partial N_j/\partial z)]$$

By using the same Gaussian weighting scheme used in subroutine FCAL, Eq. (46) may be written in the form

$$k_{ij}^e = \sum_k^9 W_k \sum_\ell^n N_\ell k_\ell \sum_i^n \sum_j^n [(\partial N_i/\partial r)(\partial N_j/\partial r) + (\partial N_i/\partial z)(\partial N_j/\partial z)]|J|$$

The resulting 8x8 elemental matrix is then added to the global stiffness matrix through the connectivity relationships.

Up to this point [K] and {f} have been assembled without regard to the boundary conditions except at the exit plane where the $(\partial\psi/\partial n) = 0$ was enforced explicitly during the procedures used by Subroutine DIST. Care must be taken to ensure that the boundary conditions for the other three segments of the boundary are not violated. As shown in section III.B, the boundary condition for the nodes along the shroud, along the hub and at the inlet plane of the machine is that the value of ψ is specified. If the value of ψ is specified at these locations the Eq. (32) must be modified so that ψ is no longer free at these nodes. A standard technique is

employed to remove individual equations from a system of equations when the degree of freedom represented by the individual equations has been removed.

The following is the subroutine's algorithm in outline form:

Algorithm:

Iterate for each element in the mesh.

Iterate for each Gaussian point.

Find shape functions, $|J|$, and $[J]^{-1}$.

Find the value of k at the Gauss point.

Compute the elemental stiffness matrix.

Upon completion of the Gaussian integration, add the local contribution to the global system.

Upon completion of the addition of the last element's contribution, modify the system of equations to include the boundary conditions.

END

12. <u>Subroutine DSIMQ</u>

Subroutine DSIMQ is a non-IMSL library, double precision subroutine that solves a set of n simultaneous equations of the form

$$[A]\{X\} = \{B\}$$

where $[A]$ is an nxn matrix

$\{X\}$ and $\{B\}$ are nx1 vectors.

13. <u>Subroutine REPLA</u>

Subroutine REPLA places the solution vector obtained from subroutine DSIMQ into its proper storage location.

14. Subroutine TEST

Subroutine TEST determines the maximum difference in the assumed nodal distribution of the stream function at the beginning of an iteration to the solution of the radial equilibrium equation calculated using the assumed distribution. The difference in the distributions at a node is defined as

$$\text{Diff} = \left| \left[\left(\psi_i^{\,n} - \tilde{\psi}_i^{\,n+1} \right) \Big/ \tilde{\psi}_i^{\,n+1} \right] \right|$$

where $\psi_i^{\,n}$ is the assumed value at node i and $\tilde{\psi}^{n+1}$ is the calculated value for node i. Convergence is considered to be reached when the maximum difference for any node is less than a selected reference value, ε.

15. Subroutine RELAX

Subroutine RELAX performs the relaxation scheme to obtain an updated estimate of the stream function distribution for the next program iteration. The new estimate for stream function distribution is calculated as follows:

$$\psi_i^{\,n+1} = \psi_i^{\,n} + \alpha \left[\tilde{\psi}_i^{\,n+1} - \psi_i^{\,n} \right] \tag{48}$$

16. Subroutine NOCON

Subroutine NOCON prepares the program for the next iteration by setting the elements of the right-hand side vector, {f}, and the stiffness matrix, [K], equal to zero.

17. Subroutine OUTPUT

Subroutine OUTPUT prints the computed nodal values of a majority of the velocities and thermodynamic properties. A listing of the values that are printed and the corresponding units is contained in Appendix D. A sample output listing is contained in Appendix G.

18. Subroutine MPLOT

Subroutine MPLOT uses the Tektronix 618 terminal to make an online graphical presentation of selected variables at the rotor inlet, rotor outlet, stator inlet and the stator outlet. Figures 7 through 20 provide examples of the plots available for display to the individual on request. The user is given the option of terminating the plotting sequence at any stage of the presentation through the use of interactive prompts.

The following is the subroutine's algorithm in outline form:

Algorithm:

Convert the appropriate variables to Real 4 for compatibility with the library plotting package GRAFF.

Determine the values of axial velocity, relative flow angles, total-to-total pressure ratio, and the adiabatic efficiency for the rotor inlet, display if requested.

Determine the values of axial velocity, and relative, absolute, and deviation angles for the rotor exit, display if requested.

Determine the values of axial velocity, absolute flow angles, and total-to-total pressure ratio for the stator inlet, display if requested.

Determine the values of axial velocity, and absolute and deviation angles for the stator exit, display if requested.

END

19. Subroutine SHAPE

Subroutine SHAPE calculates the eight nodal shape functions for a given point (ξ,η). The equations for the nodal shape functions are:

$$N(1) = (\xi\eta + \xi^2 + \eta^2 + \xi^2\eta + \xi\eta^2 - 1)/4$$

$$N(2) = (1 + \eta - \xi^2 - \xi^2\eta)/2$$

$$N(3) = (-\xi\eta + \xi^2 + \eta^2 + \xi^2\eta - \xi\eta^2 - 1)/4$$

$$N(4) = (1 - \eta^2 - \xi + \xi\eta^2)/2$$

$$N(5) = (\xi\eta + \xi^2 + \eta^2 - \xi^2\eta - \xi\eta^2 - 1)/4$$

$$N(6) = (1 - \eta - \xi^2 + \xi\eta^2)/2$$

$$N(7) = (-\xi\eta + \xi^2 + \eta^2 - \xi^2\eta + \xi\eta^2 - 1)/4$$

$$N(8) = (1 - \eta^2 + \xi - \xi\eta^2 - 1)/2$$

The values of the shape functions are stored in the array SF, and are returned to the calling portion of the program.

20. Subroutine JACOB

Subroutine JACOB computes the partial derivatives of the shape functions with respect to ξ and η and computes the elements of the Jacobian matrix, [J], for a specific point (ξ,η). The equations for the partial derivatives were obtained directly from the differentiation of the functions shown in the description of subroutine SHAPE. The arrays D

and E store the values of the $(\partial N/\partial \xi)$ and $(\partial N/\partial \eta)$ respectively. The Jacobian matrix is calculated by the following sequence of equations:

$$J(1,1) = (\partial z/\partial \xi) = \sum_i^n (\partial N_i/\partial \xi) z_i$$

$$J(1,2) = (\partial r/\partial \xi) = \sum_i^n (\partial N_i/\partial \xi) r_i$$

$$J(2,1) = (\partial z/\partial \eta) = \sum_i^n (\partial N_i/\partial \eta) z_i$$

$$J(2,2) = (\partial r/\partial \eta) = \sum_i^n (\partial N_i/\partial \eta) r_i$$

Arrays D and E and the Jacobian matrix are returned to the calling location in the program.

21. <u>Subroutine ERR1</u>

Subroutine ERR1 is called by subroutine INPUT if the storage limitation for Real 8 variables has been exceeded. The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The user response would be to increase the value of LIMR if possible or reduce the size of the mesh.

22. <u>Subroutine ERR2</u>

Subroutine ERR2 is called by subroutine INPUT if the storage limitation for Real 4 variables has been exceeded. The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The

user response would be to increase the value of LIM4 if possible or reduce the size of the mesh.

23. Subroutine ERR3

Subroutine ERR3 is called by subroutine INPUT if the storage limitation for Integer 4 variables has been exceeded. The subroutine displays the amount by which the limitation was exceeded and terminates the program's execution. The user response would be to increase the value of LIMI if possible or reduce the size of the mesh.

VI. RESULTS AND DISCUSSION

A. PROGRAM VERIFICATION

Four operating conditions of the NASA TASK-1 compressor were used to test the capabilities of the programs MESHGEN and TURBO. In all cases the 63 element, 222 node mesh with an under-relaxation factor of 0.24 as recommended by Hirsch and Warzee [Ref. 4] were used. Selected portions of the results obtained are presented in Figs. 7 through 76. The points annotated as "observed values" were obtained from the material published in Refs. 23 and 24. The values attributed to Gavito were obtained from Ref. 8 and those attributed to Hirsch from Ref. 4. A discussion of the predictions for the various conditions are presented in the sections that follow.

1. Test Case 1

For test case 1 the operating point was defined as a rotor speed of 50% design speed and an inlet mass flow rate of 107.6 lbm/sec. The author was unable to locate this specific operating point in Ref. 23 or 24 and must assume that in Ref. 4 the mass flow rate was modified to conform to the end-wall boundary layer scheme described in that reference. Therefore, it was necessary to use the observed values published by Hirsch and Warzee in Figs. 21 through 30. The relative differences found between the present predictions and the reported observations are given in Table 1.

a. Comparison to the Work of Gavito

Figures 21, 22, 23 and 24 compare the results obtained by Gavito with the predictions of the present program. A significant improvement has been obtained at all locations. That this was possible was due in large measure to the solid foundation to the present work provided by Gavito's program and to the excellent documentation given in Ref. 8.

b. Comparison to the Work of Hirsch and Warzee

The predictions of the program TURBO compare quite faborably with those of Hirsch and Warzee. As shown in Figs. 25, 26, 29 and 30 the predictions of the two programs have almost identical average relative errors for the velocity profiles. The predictions of Hirsch and Warzee tend to have better agreement in the rotor and stator tip regions, while the program TURBO has slightly better agreement near the hub. The program TURBO's predictions had a 3.7% and 2.5% average error at the rotor inlet and exit respectively with a maximum error of 4.6% at the inlet and 7.0% at the outlet. The stator inlet velocity predictions had an average relative error of 2.6% and maximum error of 5.8% and the outlet predictions had a 2.0% average error with a maximum error of 6.0%. The prediction of both programs for the velocity profiles show excellent agreement with the observed values.

Hirsch and Warzee's program consistently produced flow angle predictions with closer agreement to the observed

values for the published rotor outlet angles. Though the two programs had the same average errors of 3.3° for the relative flow angles and 4.8° for the absolute flow angles, Hirsch and Warzee's program provided better qualitative distributions. The difference is clearly shown in Figs. 27 and 28.

Hirsch and Warzee did not publish predictions of total pressure ratios or adiabatic efficiencies, so the predictions made by TURBO for these parameters were not presented. It is assumed that no significant differences could have occurred because of the similarity of the results for the velocity profile and flow angles discussed earlier.

2. Test Case 2

Because of the apparent modification in the mass flow rate which was assumed in test case 1 and the lack of comparative data for all quantities predicted by the program TURBO, another operating point at 50% design speed was compared. The operating condition for case 2 was defined as a mass flow rate of 114.7 lbm/sec at a speed equal to 50% of design, which corresponded to reading 38 of Ref. 24. The results for case 2 are presented in Figs. 31 through 44, and a summary of the relative differences between the predictions and observations is contained in Table 2.

3. Test Case 3

Test case 3 corresponded to reading 45 of Ref. 24, which was defined as a speed equal to 70% design and a mass flow rate of 151.55 lbm/sec. When reviewing the results

presented for this case, one should note the decline in the agreement between the program's predictions and the observed values. It is the author's opinion that the degradation is primarily caused by two factors. The first is the formulation used to compute the meridional velocity change across the rotor and the other is the application of a single blockage factor to all nodes to account for the end-wall boundary layers. Both factors are much more significant at 70% design speed than they were at 50%. At 70% design speed the rotor tip relative Mach number is about 0.94. This would require the program to account for transonic effects at the tip. Second, the mass flow and absolute Mach number of the flow in all regions is significantly higher at 70% design speed. Therefore, it is unlikely that a single blockage factor will work satisfactorily in all regions of the machine. It is hoped that both areas will be addressed in any future work on the program.

The results for test case 3 are presented in Figs. 45 through 58 with the corresponding differences between the predictions and observations summarized in Table 3.

4. Test Case 4

The operating condition of test case 4 was at 80% design speed point with a mass flow rate of 174.54 lbm/sec, corresponding to reading 50 of Ref. 24. Figures 59 through 72 and Table 4 present the results of the program's predictions and comparisons to the observed values. As expected,

and for similar reasons to those cited in test case 3, the agreement between the program's predictions and observed values is significantly poorer than any of the three previous cases.

B. POINTS OF INTEREST

The results produced by the program are highly dependent on the value of the general blockage factor used to account for the end-wall boundary layers. This factor influences both the quality, in terms of agreement with observations, and the stability of the solution. Figures 73 through 76 show the axial velocity distributions for the rotor and stator for test case 4 with a blockage factor of 9% instead of the 6% factor used to obtain the results shown in Figs. 59 through 72. A comparison of corresponding velocity profiles clearly demonstrates the factor's pronounced influence on the program's solution. The general blockage factor also has a strong influence on the program's convergence rate. In some cases the factor can cause the program to become oscillatory or even divergent.

For the low speed cases of 50% and 70% design, additional blockage factors had to be applied to the duct element between the rotor and stator tips to obtain accurate results. As shown in the program listing for subroutine INPUT of program TURBO, the factors used for 50% design speed were much higher than the factors used for 70% design speed and that

no additional factors were used for 80% design speed. A global method of calculating the end-wall blockage factor must be incorporated if the program is to become independent of inputs other than physical constants.

The deterioration of the program's predictions with increasing Mach number and its failure to run for test cases with strong supersonic relative velocities at the tip demonstrate the need to provide the program with a method of handling supersonic relative velocities. Hirsch and Warzee [Ref. 15] showed a method of extending the radial equilibrium formulation used in the present code to supersonic flow. They presented comparisons of predictions obtained by this method to observations of the NASA TASK-1 transonic compressor at 100% design speed. The results were impressive and clearly showed that the method is valid for relative velocities in excess of Mach 1.4. It is strongly recommended that the first effort at improving the code be an effort to modify TURBO to include the method shown in Ref. 15.

VII. CONCLUSIONS AND RECOMMENDATIONS

A computer program based on the finite element technique has been developed and has been verified satisfactorily for computing flows through subsonic axial flow compressor stages. Minor modifications have been suggested to allow transonic stages to be calculated.

The code was written in such a way that it could be readily adapted to compute either turbines or compressors with multiple stages. Before such extensions are attempted however, the following specific recommendations are made to improve the present compressor code.

A. PROGRAM MESHGEN

1. Incorporate some of the two-dimensional techniques of Adamek [Ref. 25] to improve the efficiency of the code.
2. Review the code to find improvements in storage allocations and computational efficiencies.
3. Modify the program to track the first and last nodes of the rotor and stator as subscripted variables so that the program can be used to generate the appropriate mesh parameters for a multi-stage machine.
4. Convert subroutine MPLOT to the DISPLA system.

B. PROGRAM TURBO

1. Incorporate a method for the global calculation of the blockage and losses created by the end-wall boundary layers.

2. Convert the storage of [K] and the solution technique for the program to at least a symmetric banded scheme or if possible to a skyline equivalent scheme.

3. Test the program on a variety of machines and operating conditions.

4. Obtain expressions that approximate the NASA correlation curves for 65-series blading.

5. Incorporate methods for the prediction of stall/surge.

6. Take advantage of the modular form of the program and include a variety of correlation techniques as a user selected option.

7. Review the program for improved storage and computational techniques. Specifically, determine ways to take fuller advantage of the dynamic dimensioning scheme [Ref. 16] used by the program.

8. Convert the rotor inlet calculations to allow the value of β_1 to have nonzero values so that the program can be extended to multi-stage analysis.

9. Modify the use of the values of the beginning nodes of the rotor and stator to subscripted variables so that the analysis can be extended to multi-stage machines.

10. Convert subroutine MPLOT to the DISPLA system.

11. Develop iterative schemes for calculating the flow angle, the velocity distribution changes and the thermodynamic property changes across a turbine rotor and stator for inclusion in subroutines ROTO and STAT.

LIST OF REFERENCES

1. Naval Postgraduate School Technical Report NPS-57MA70081A, Computer Program for Prediction of Axial Flow Turbine Performance, by E. Macchi, August 1970.

2. Cirone, R., Computer Evaluation of the On- and Off-Design Performance of an Axial Air Turbine, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1981.

3. Naval Postgraduate School TPL Technical Note 82-02, Verification of a Single Stage Axial Turbine Performance Prediction Program for the HP 21-MX Computer System, by J. A. Ferguson, May 1982.

4. Hirsch, C., and Warzee, G., "A Finite Element Method for the Axisymmetric Flow Computation in a Turbomachine," International Journal for Numerical Methods in Engineering, v. 10, pp. 93-113, 1976.

5. Adler, D., and Krimerman, Y., "The Numerical Calculation of the Meridional Flow Field in Turbomachines Using the Finite Element Method," Israel Journal of Technology, v. 12, pp. 268-274, 1974.

6. Hirsch, C., and Deconinck, H., Finite Element Methods for Transonic Blade-to-Blade Calculation in Turbomachines, ASME Paper 81-GT-5.

7. Bettencourt, J., Finite Element Analysis Program (FEAP) for Conduction Heat Transfer, Engineer's Thesis, Naval Postgraduate School, Monterey, California, December 1979.

8. Gavito, F. V., The Finite Element Method Applied to Flows in Turbomachines, M.S. Thesis, Naval Postgraduate School, Monterey, California, December 1976.

9. NACA TN2604, A General Theory of Three-dimensional Flow in Subsonic and Supersonic Turbomachines of Axial, Radial, and Mixed Flow Types, by C. H. Wu, 1952.

10. Smith, H. L., "The Radial Equilibrium Equation of Turbomachinery," Transactions of the ASME, Journal of Engineering for Power, v. 88A, pp. 1-12, 1966.

11. Novak, R. A., "Streamline Curvature Computing Procedures for Fluid Flow Problems," Transactions of the ASME, Journal of Engineering for Power, v. 89A, p. 478, 1967.

12. Aeronautical Research Council R&M 3509, A Digital Computer Program for the Through-flow Fluid Mechanics in an Arbitrary Turbomachine Using a Matrix Method, by H. Marsh, 1966.

13. Carleton University, Division of Aerothermodynamics, ME/A 73-1, Axial Flow Compressor Analysis Using a Matrix Method, by W. R. Davis and D. A. J. Miller, 1973.

14. Vavra, M. H., Aero-Thermodynamics and Flow in Turbomachines, Kriger, 1981.

15. Hirsch, C., and Warzee, G., "A Finite Element Method for Through-flow Calculations in Turbomachines," Transactions of the ASME, Journal of Fluids Engineering, v. 98, pp. 403-422, 1976.

16. Cook, R. D., Concepts and Applications of Finite Element Analysis, pp. 2, 399-401, Wiley, 1981.

17. Huebner, K. H., The Finite Element Method for Engineers, Wiley, 1975.

18. Zienkiewicz, O. C., The Finite Element Method, 3rd Ed., McGraw-Hill, 1977.

19. Kaplan, W., Advanced Calculus, p. 93, Addison-Wesley, 1952.

20. Naval Postgraduate School Technical Report NPS-57Sf74081, Flow Into a Transonic Compressor Rotor, Part 1, Analysis, by R. P. Shreeve, August 1974.

21. NASA SP-36, Aerodynamic Design of Axial Flow Compressors, by NASA Staff, 1965.

22. Okiishi, T. H., private communication, June 1982.

23. NASA CR-72806, Evaluation of Range and Distortion Tolerance for High Mach Number Transonic Fan Stages, Volume 1, by C. C. Koch, K. R. Bilwakesh, and V. L. Doyle, August 1971.

24. NASA CR-72964, Evaluation of Range and Distortion Tolerance for High Mach Number Transonic Fan Stages, Volume 2, by C. C. Koch, K. R. Bilwakesh, and V. L. Doyle, August 1971.

25. Adamek, J. M. An Automatic Mesh Generator Using Two- and Three-dimensional Isoparametric Finite Elements, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1973.

TABLE 1

Connectivity Relationships for Figure Three

Element Number	Local Node Number	Global Number
1	1	12
1	2	8
1	3	1
1	4	2
1	5	3
1	6	9
1	7	14
1	8	13
2	1	14
2	2	9
2	3	3
2	4	4
2	5	5
2	6	10
2	7	16
2	8	15
3	1	16
3	2	10
3	3	5
3	4	6
3	5	7
3	6	11
3	7	18
3	8	17

TABLE 2

Comparison of Program Predictions with NASA Task-1 Compressor Measurements at 50% Design Speed

	Average Difference	Maximum Difference
Rotor Inlet		
Axial Velocity	4.6%	7.2%
Relative Angles	1.5°	3.1°
Total Pressure Ratio	0.7%	1.4%
Efficiencies	4.3%	11.4%
Rotor Outlet		
Axial Velocity	3.2%	6.7%
Relative Angles	1.6°	2.8°
Absolute Angles	1.5°	4.2°
Deviation Angles	2.2°	3.0°
Stator Inlet		
Axial Velocity	3.4%	6.8%
Absolute Angles	2.5°	4.2°
Total Pressure Ratio	0.3%	1.2%
Stator Outlet		
Axial Velocity	1.7%	5.7%
Absolute Angles	0.6°	1.5°
Deviation Angles	1.4°	2.6°

TABLE 3

Comparison of Program Predictions with NASA Task-1 Compressor Measurements at 70% Design Speed

	Average Difference	Maximum Difference
Rotor Inlet		
Axial Velocity	4.8%	7.4%
Relative Angles	1.14°	1.8°
Total Pressure Ratio	1.1%	2.0%
Efficiencies	3.2%	4.8%
Rotor Outlet		
Axial Velocity	4.1%	7.4%
Relative Angles	2.0°	3.2°
Absolute Angles	3.6°	5.5°
Deviation Angles	1.6°	3.5°
Stator Inlet		
Axial Velocity	4.0%	8.8%
Absolute Angles	3.9°	5.5°
Total Pressure Ratio	0.5%	2.1%
Stator Outlet		
Axial Velocity	3.6%	7.4%
Absolute Angles	0.6°	1.1°
Deviation Angles	1.5°	2.4°

TABLE 4

Comparison of Program Predictions with NASA Task-1 Compressor Measurements at 80% Design Speed

	Average Difference	Maximum Difference
Rotor Inlet		
Axial Velocity	7.9%	11.4%
Relative Angles	1.3°	1.5°
Total Pressure Ratio	1.5%	2.5%
Efficiencies	3.1%	7.9%
Rotor Outlet		
Axial Velocity	6.6%	8.9%
Relative Angles	2.9°	5.5°
Absolute Angles	6.8°	8.5°
Deviation Angles	2.4°	4.7°
Stator Inlet		
Axial Velocity	6.9%	13.9%
Absolute Angles	5.6°	7.0°
Total Pressure Ratio	0.9%	3.1%
Stator Outlet		
Axial Velocity	8.0%	20.0%
Absolute Angles	0.8°	1.4°
Deviation Angles	1.5°	2.3°

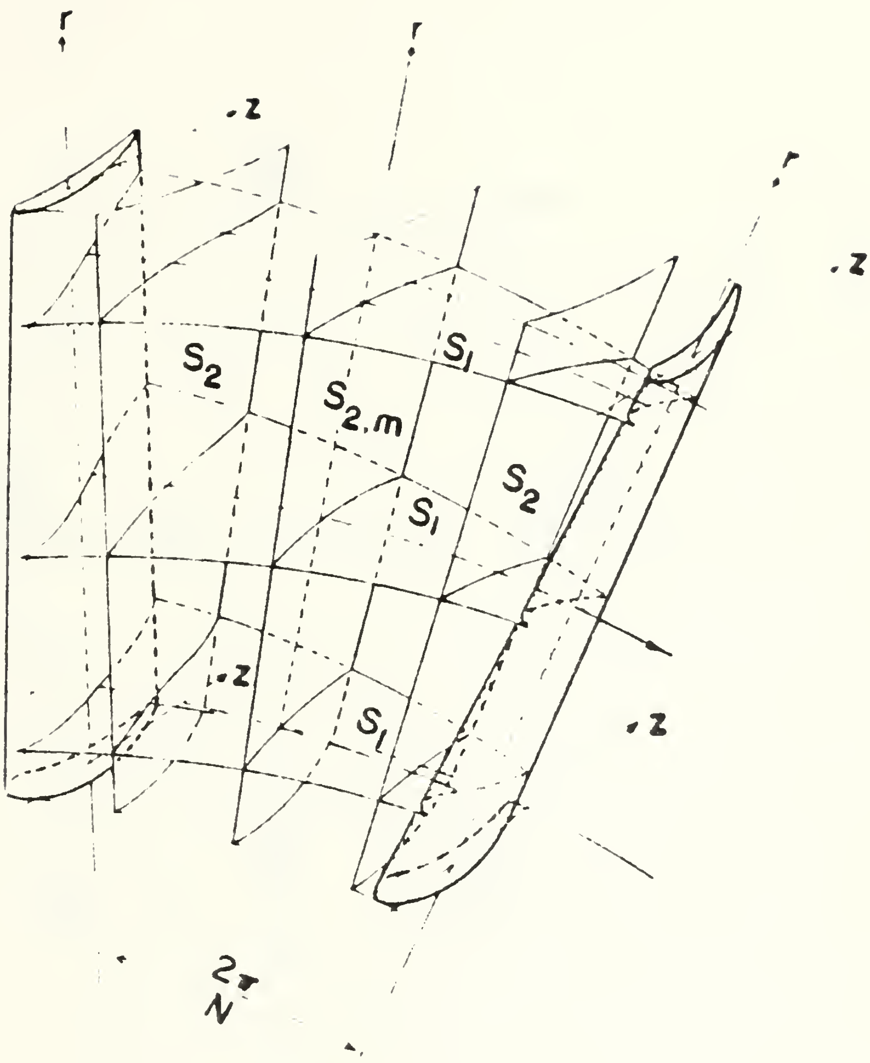

Figure 1. S1 and S2 Stream Surfaces

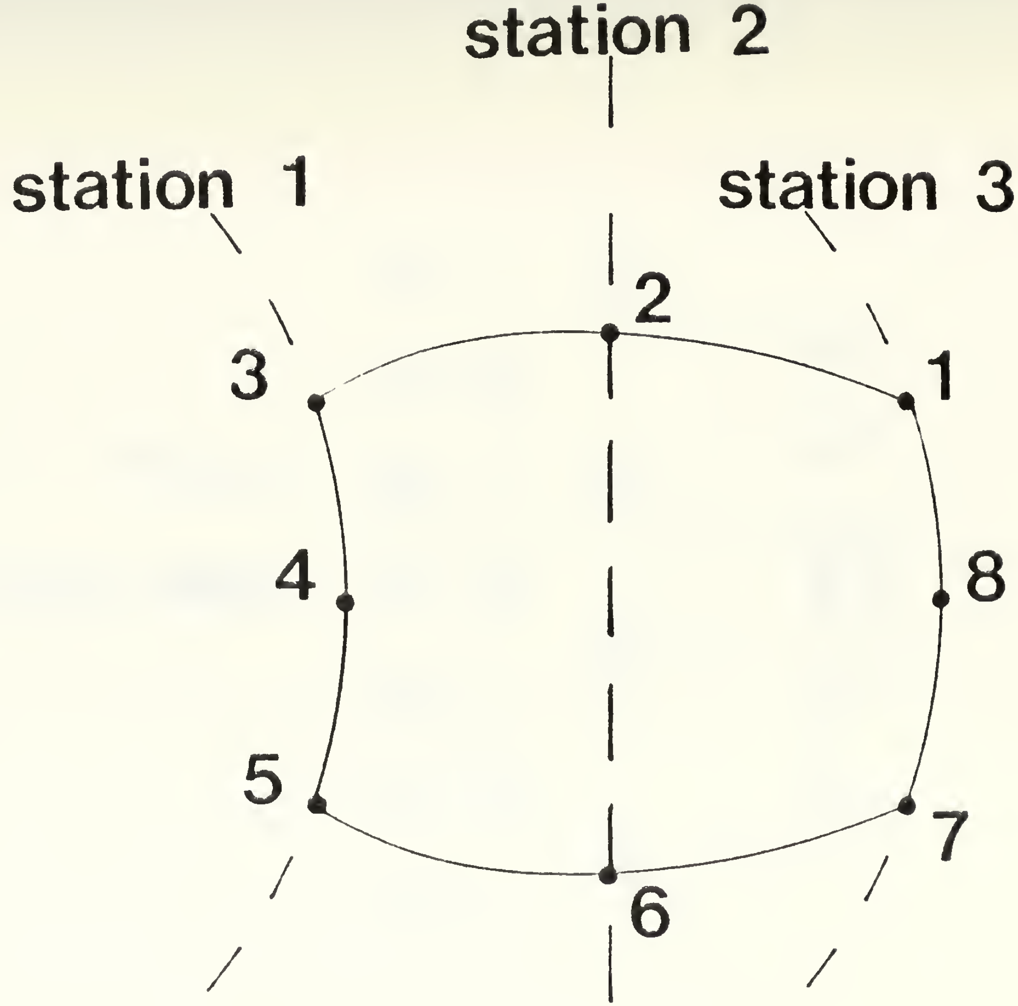

Figure 2. Nomenclature of an Eight-Node Element

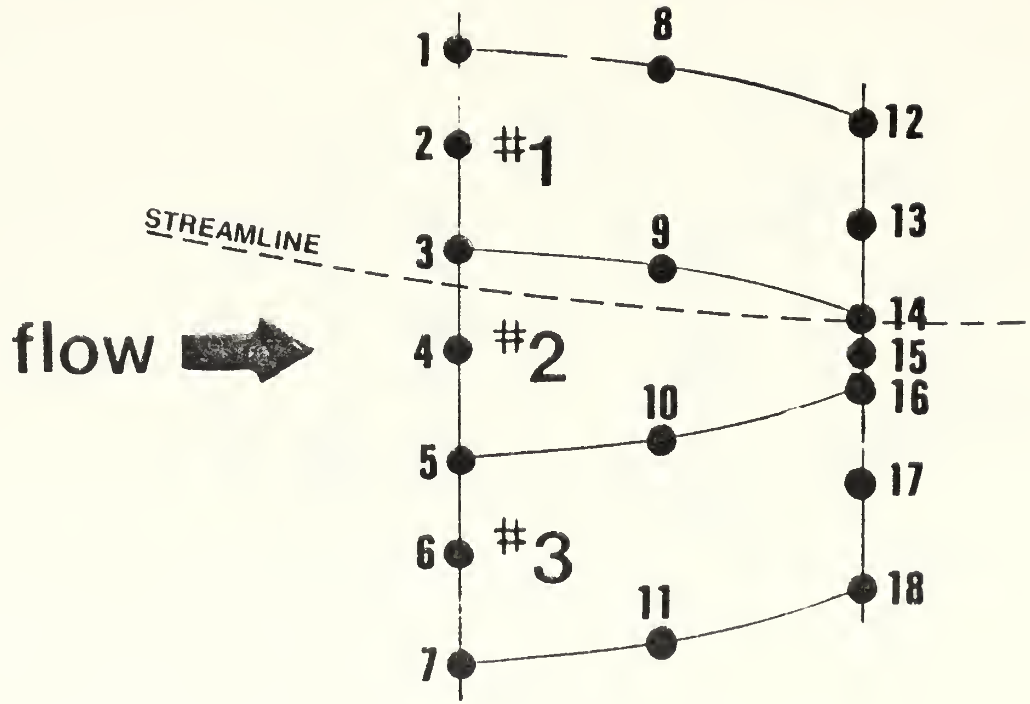

Figure 3. Example of a Three Element Mesh

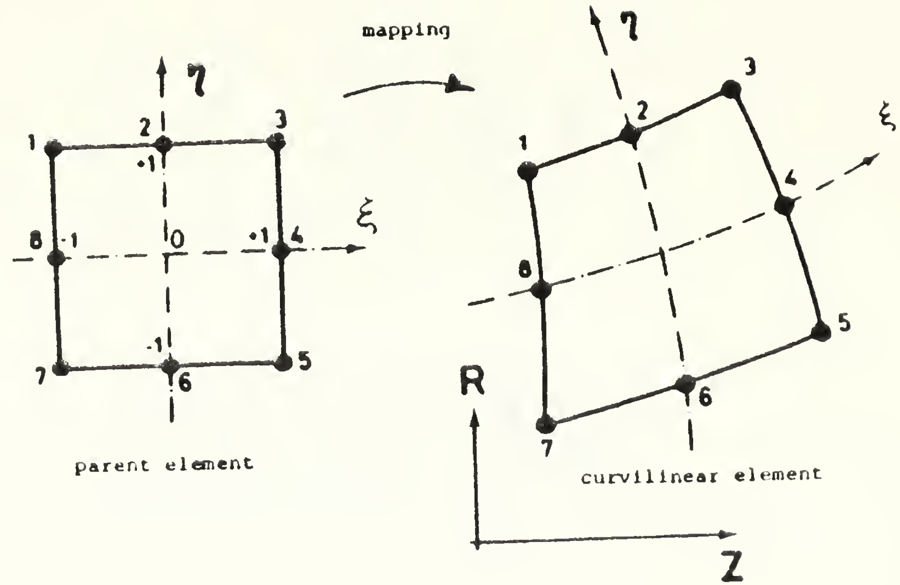

Figure 4. Mapping Relationships from the ξ,η to the z,r Plane

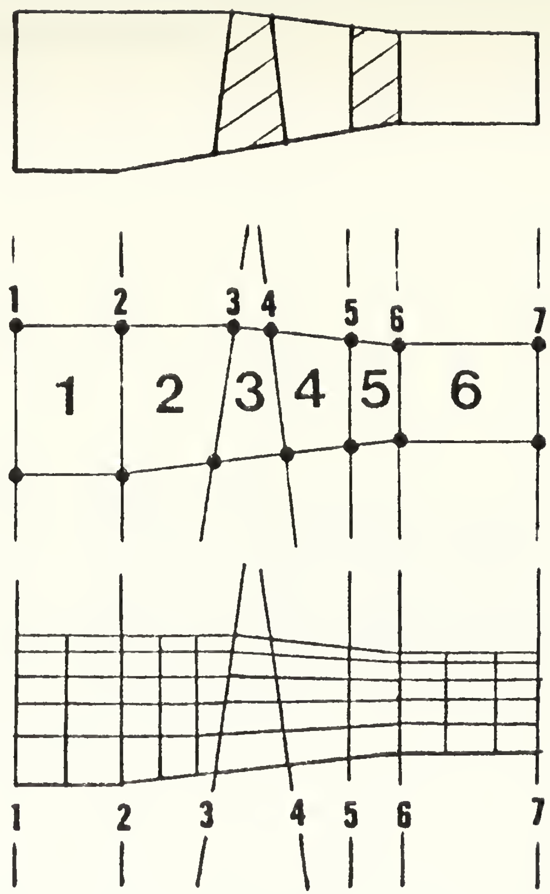

Figure 5. Mesh Generation

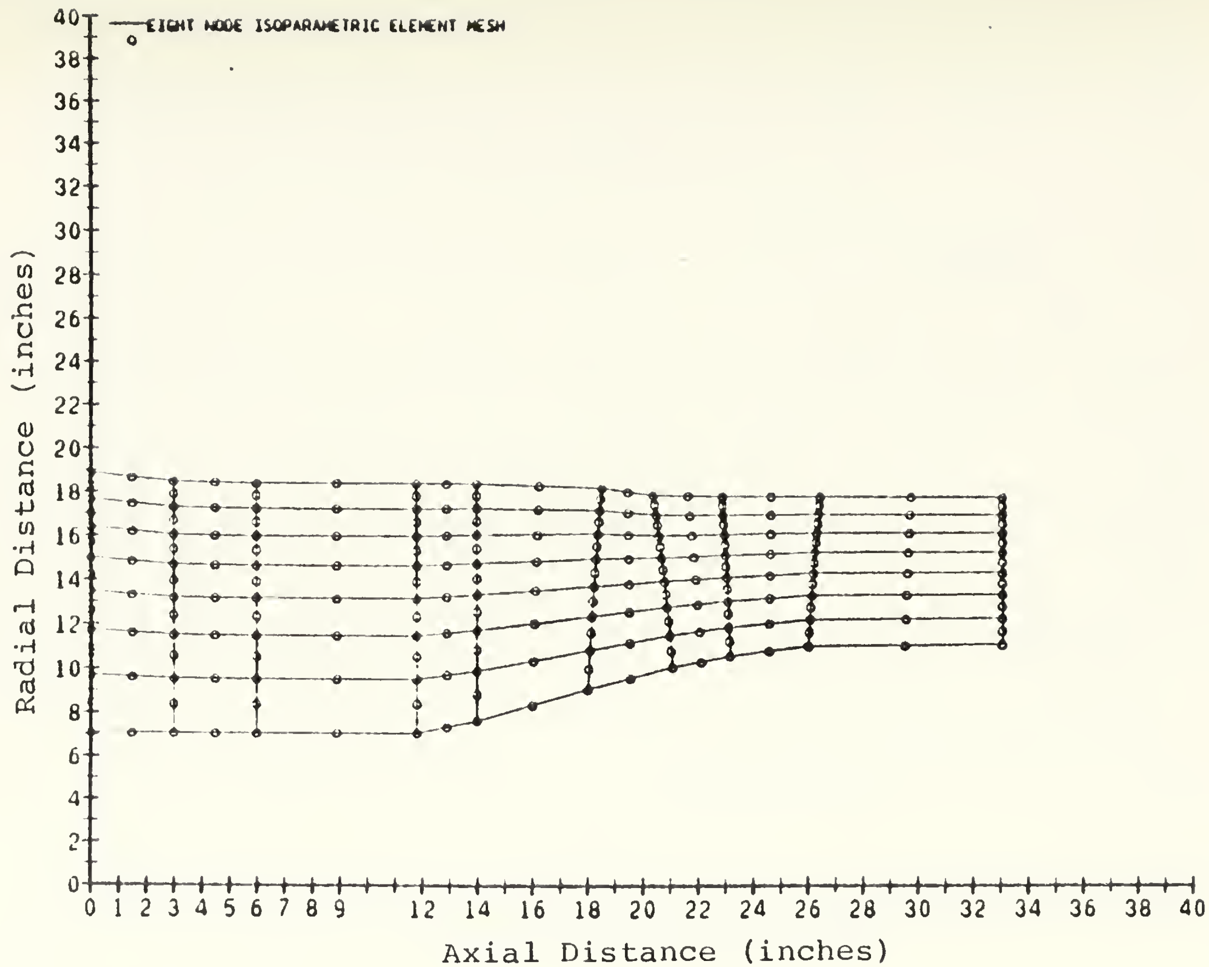

Figure 6. MESHGEN Generated Plot of the 222 Node Mesh Used in the Calculations of the Meridional Through-Flow

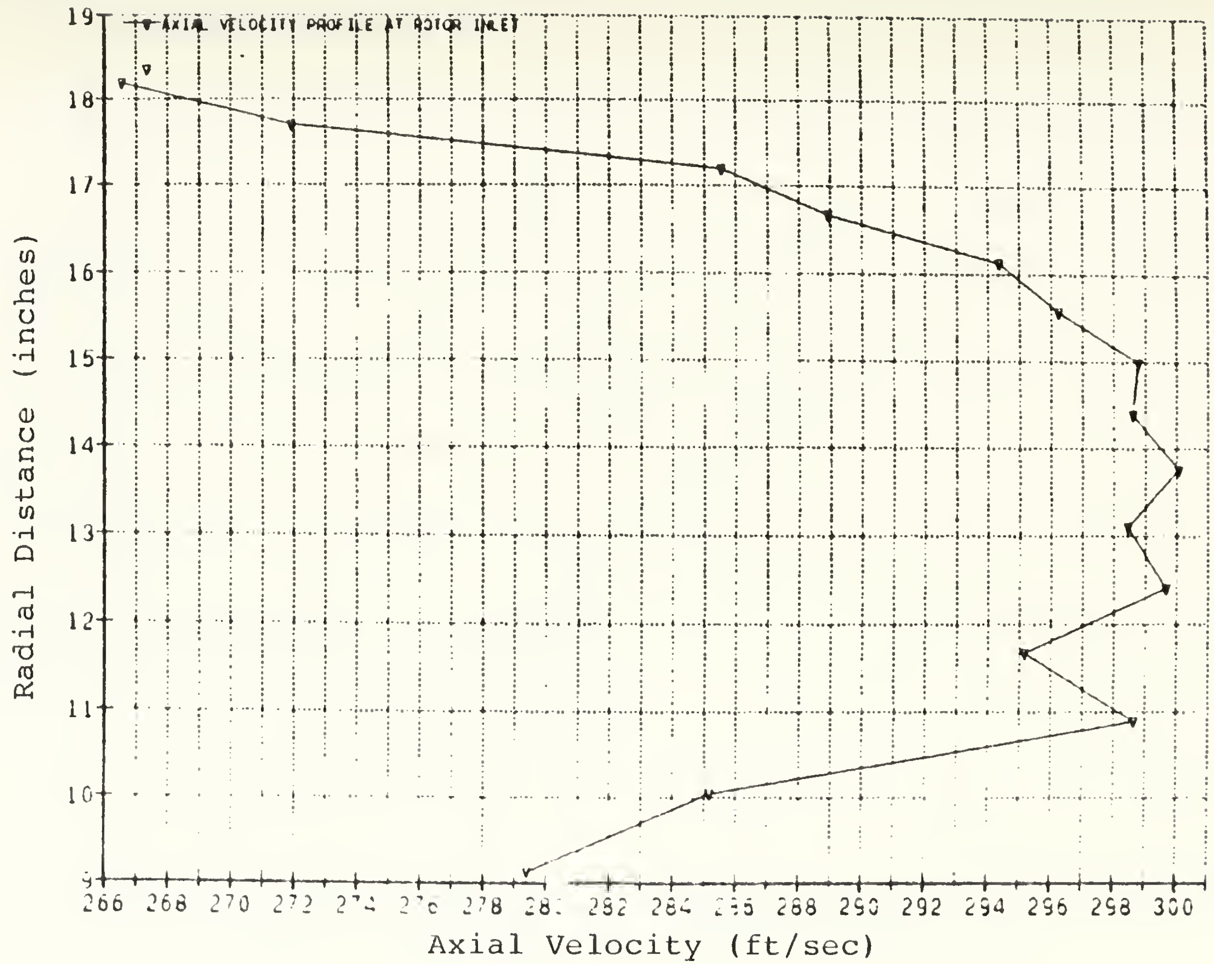

Figure 7. A TURBO Generated Tektonix 618 Plot of the Rotor Inlet Axial Velocity at 50% Design Speed

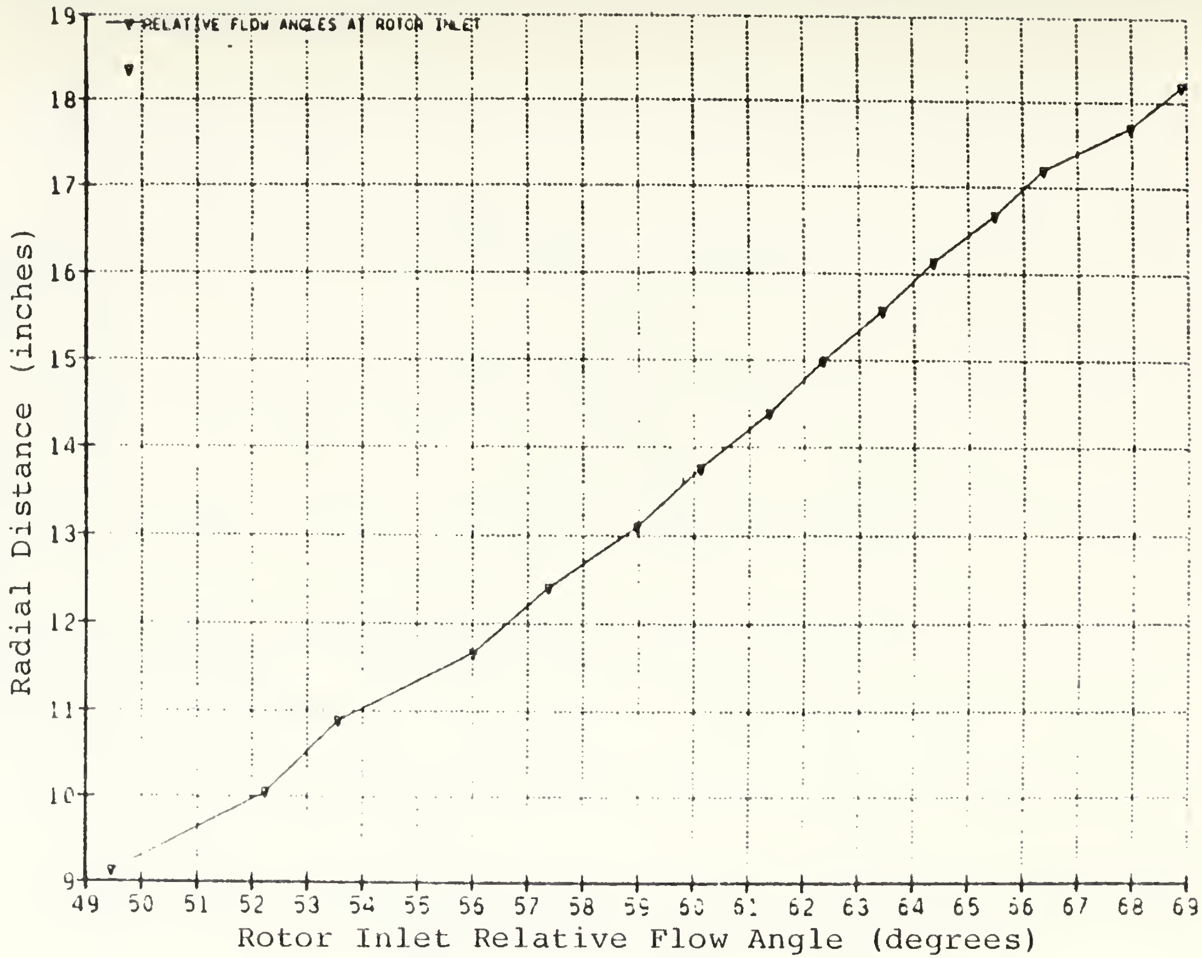

Figure 8. A TURBO Generated Tektonix 618 Plot of the Rotor Inlet Relative Flow Angles

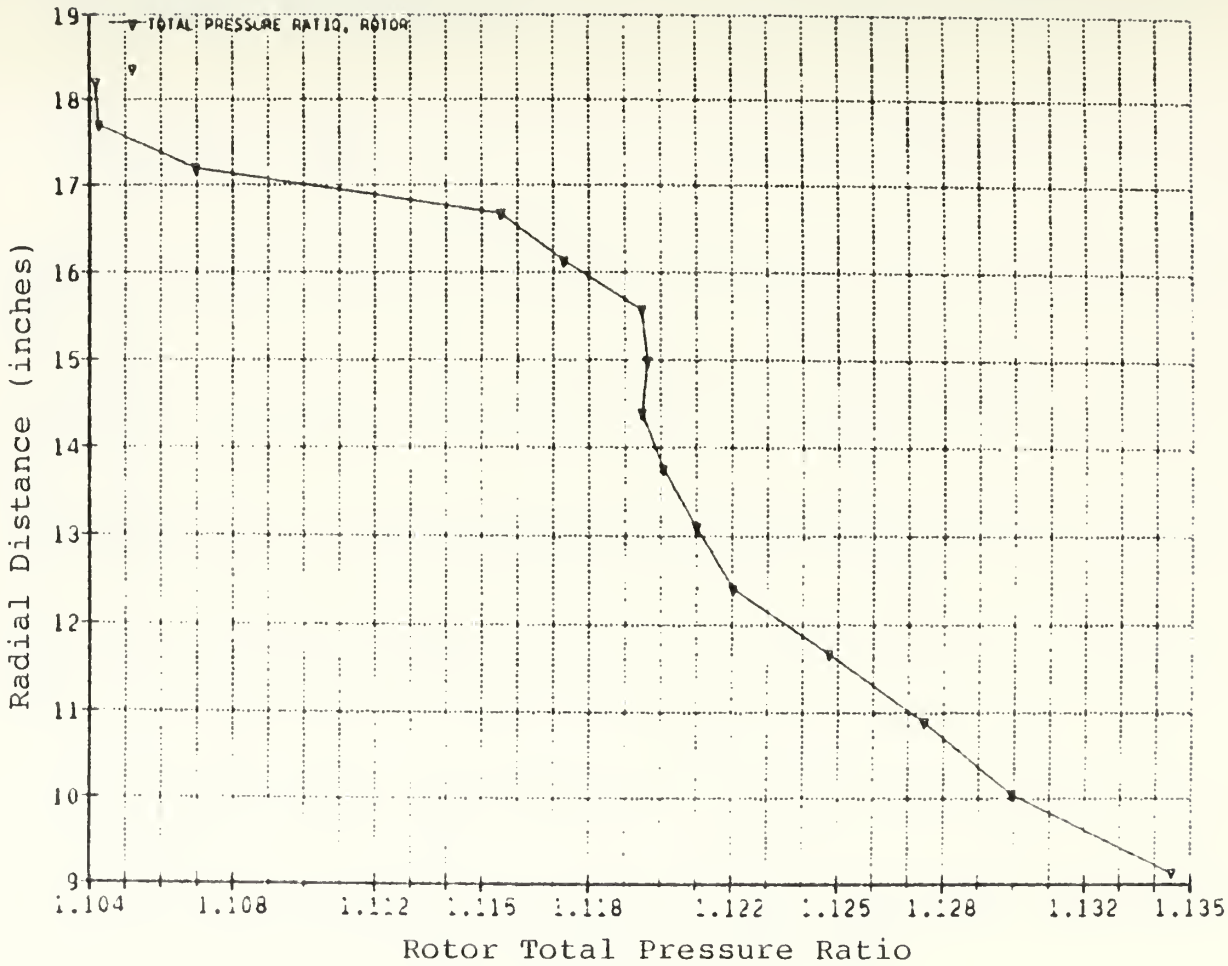

Figure 9. A TURBO Generated Tektonix 618 Plot of the Total Pressure Ratio of the Rotor vs Inlet Radius

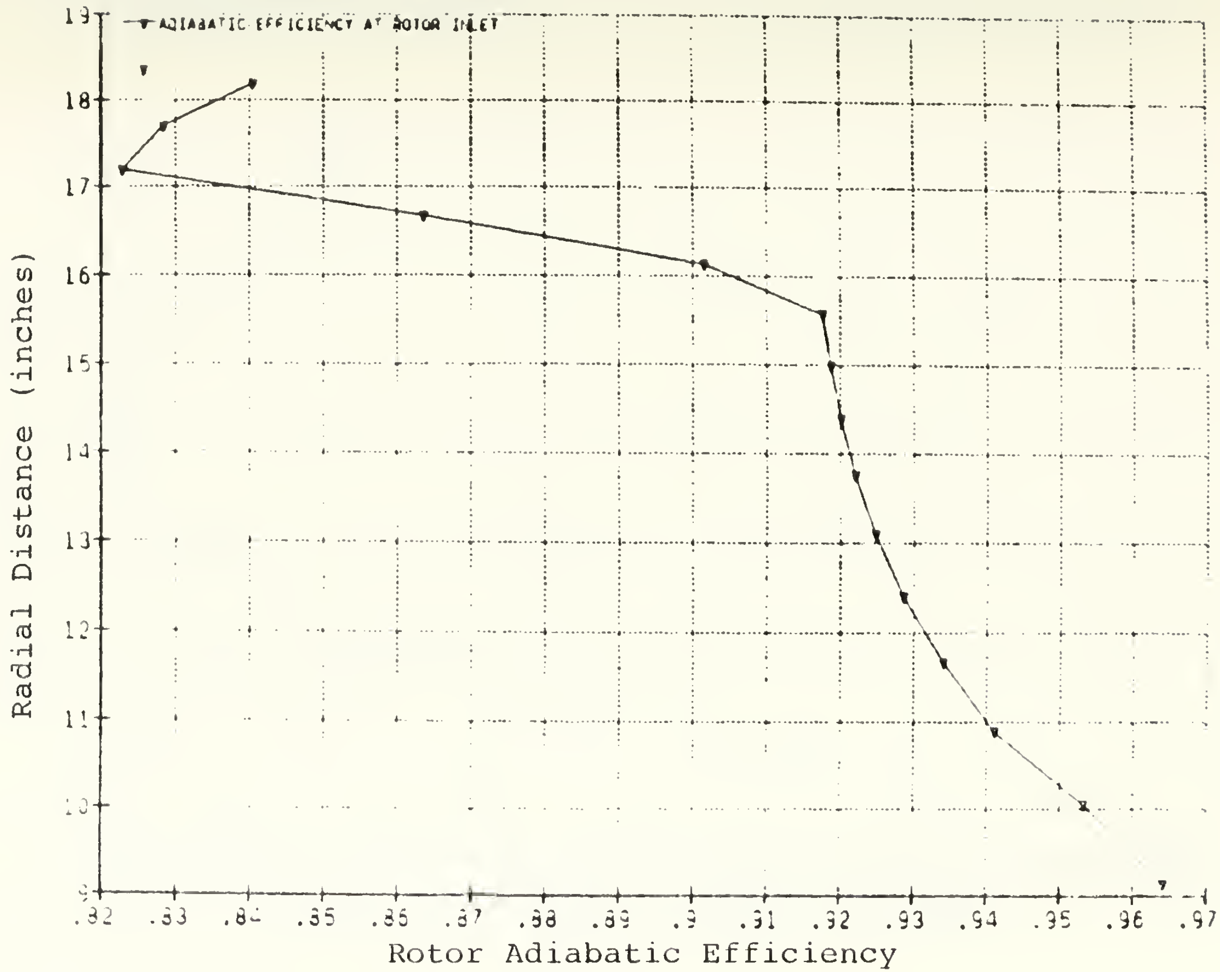

Figure 10. A TURBO Generated Tektonix 618 Plot of the Adiabatic Efficiency of the Rotor vs Inlet Radius

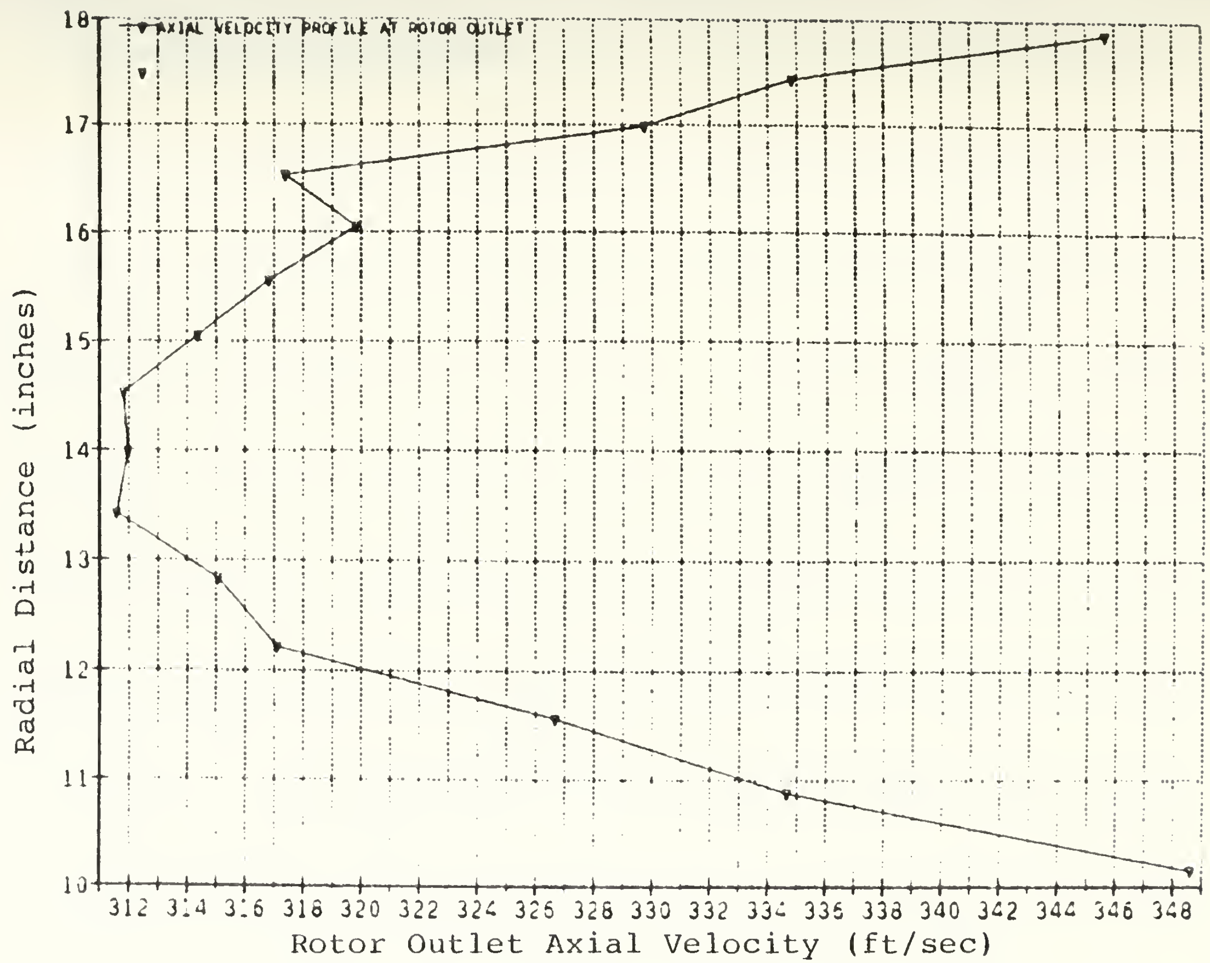

Figure 11. A TURBO Generated Tektonix 618 Plot of the Rotor Outlet Axial Velocity

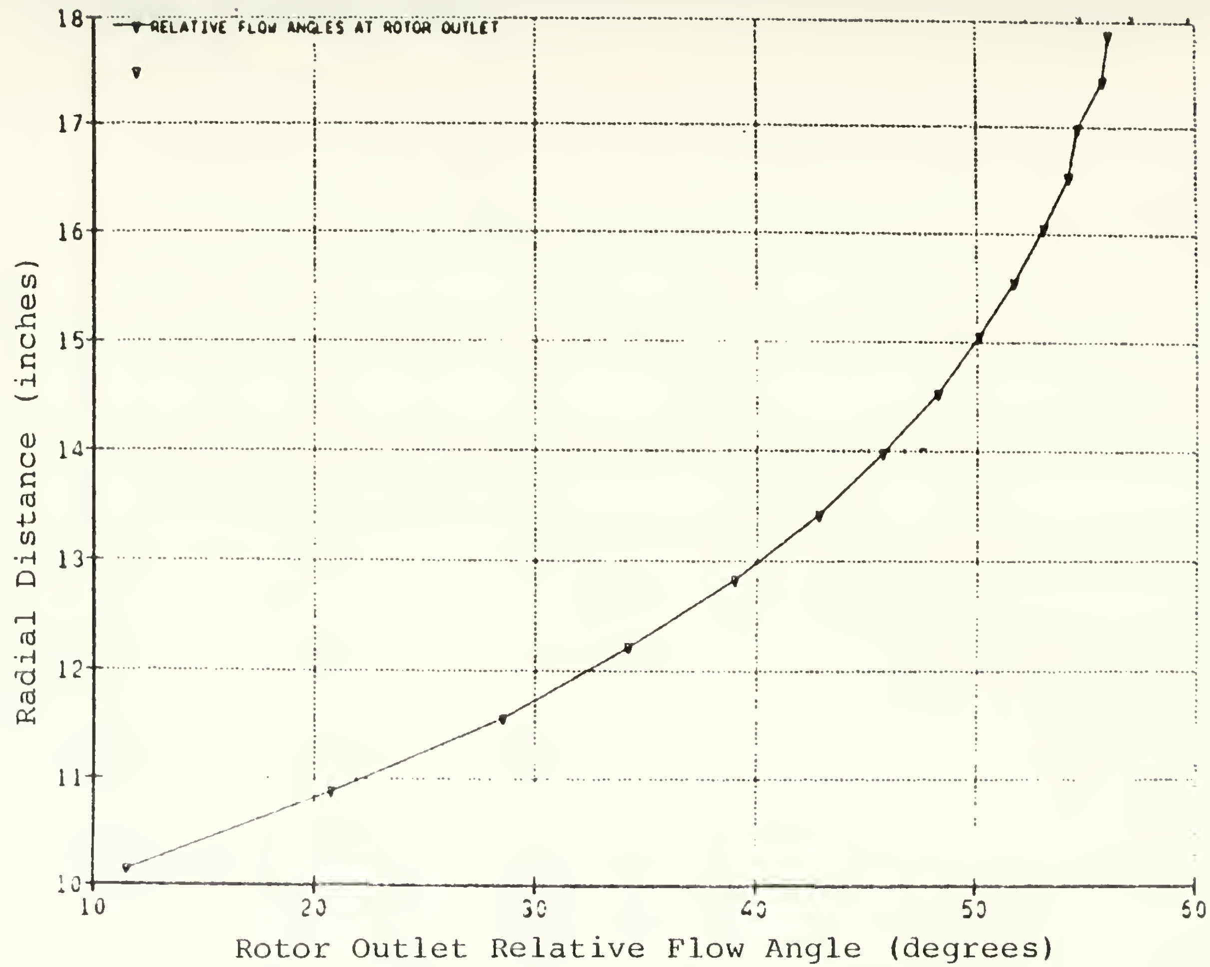

Figure 12. A TURBO Generated Tektonix 618 Plot of the Rotor Outlet Relative Flow Angles

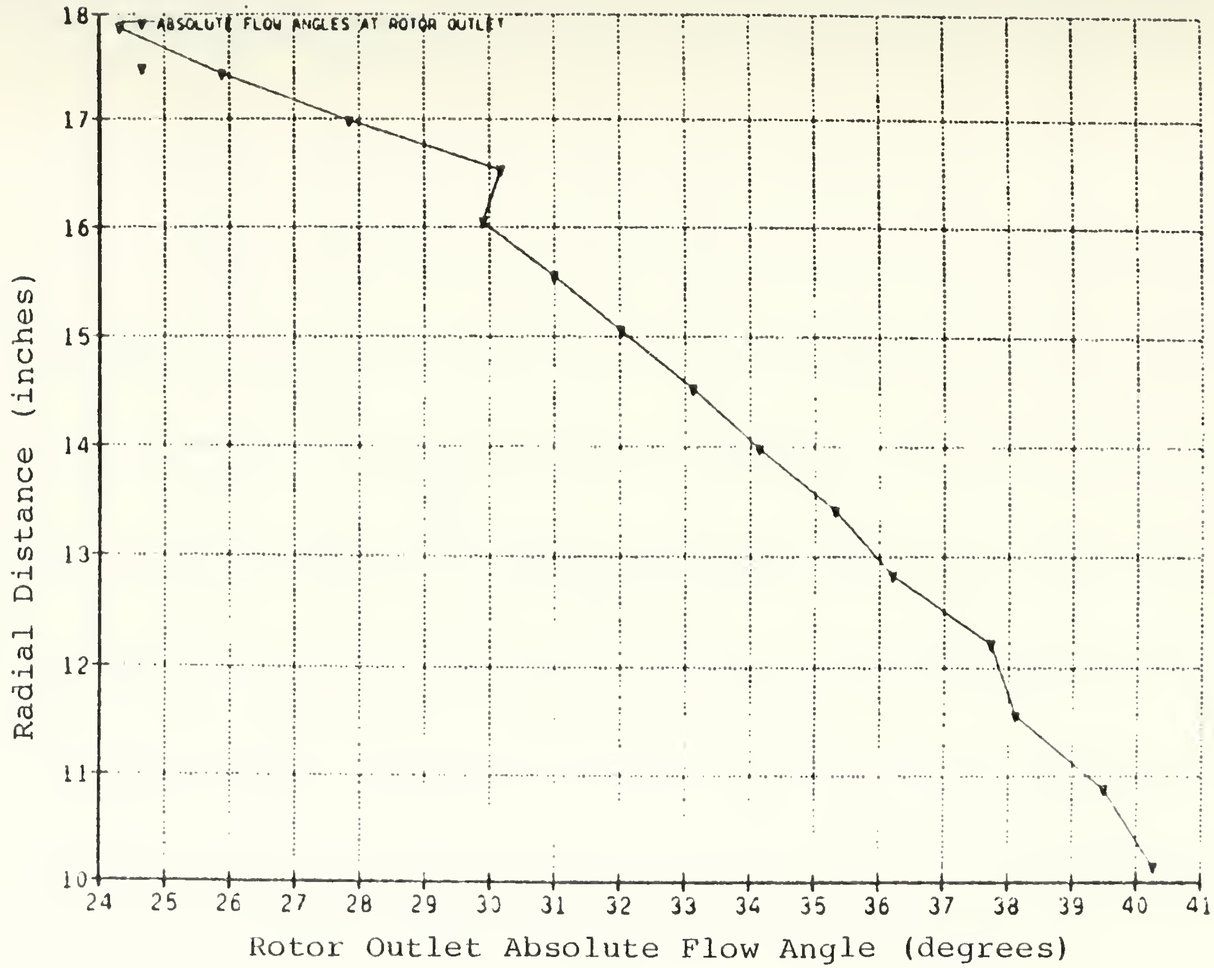

Figure 13. A TURBO Generated Tektonix 618 Plot of the Rotor Outlet Absolute Flow Angles

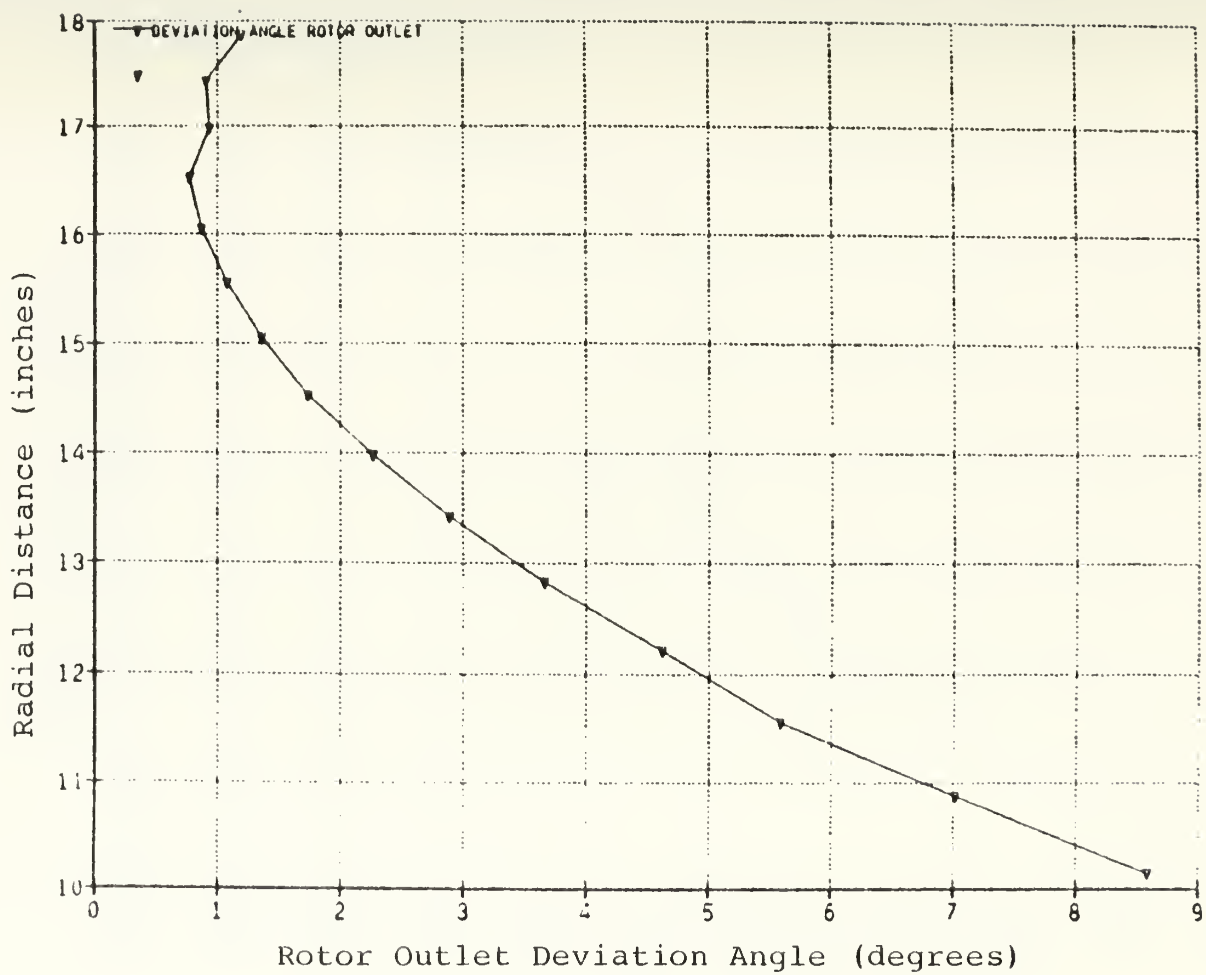

Figure 14. A TURBO Generated Tektonix 618 Plot of the Rotor Outlet Deviation Flow Angles

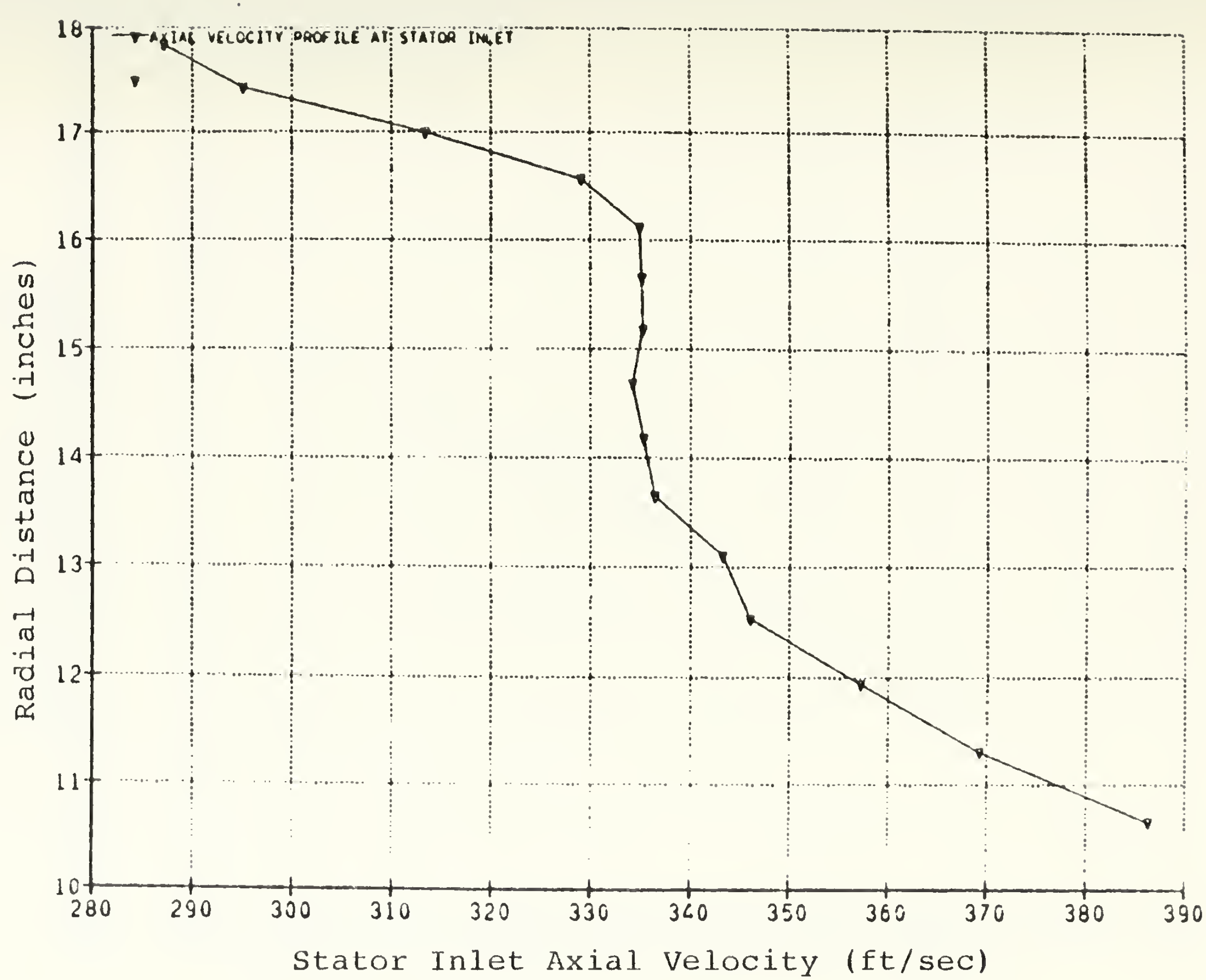

Figure 15. A TURBO Generated Tektonix 618 Plot of the Stator Inlet Axial Velocity

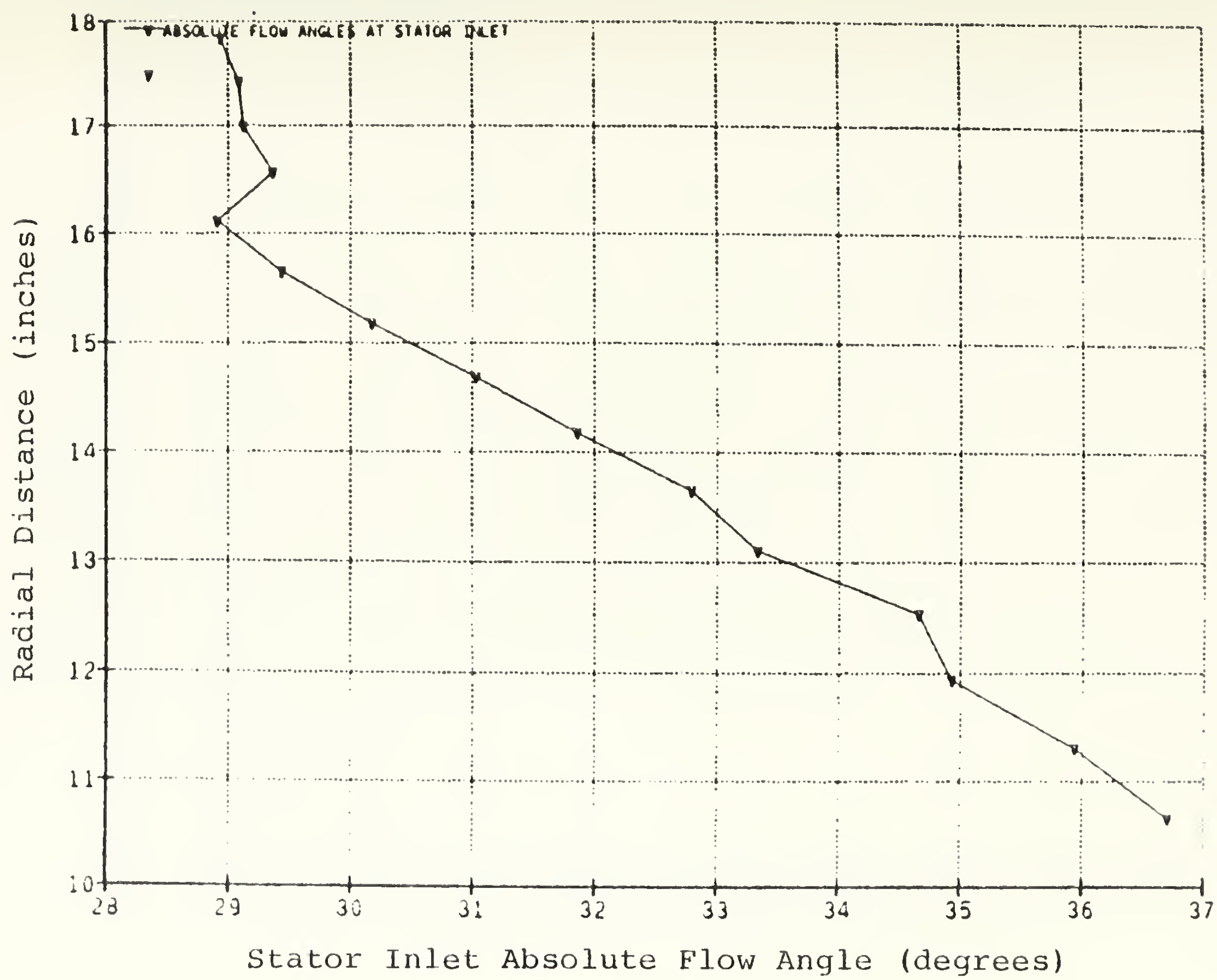

Figure 16. A TURBO Generated Tektonix 618 Plot of the Stator Inlet Absolute Flow Angles

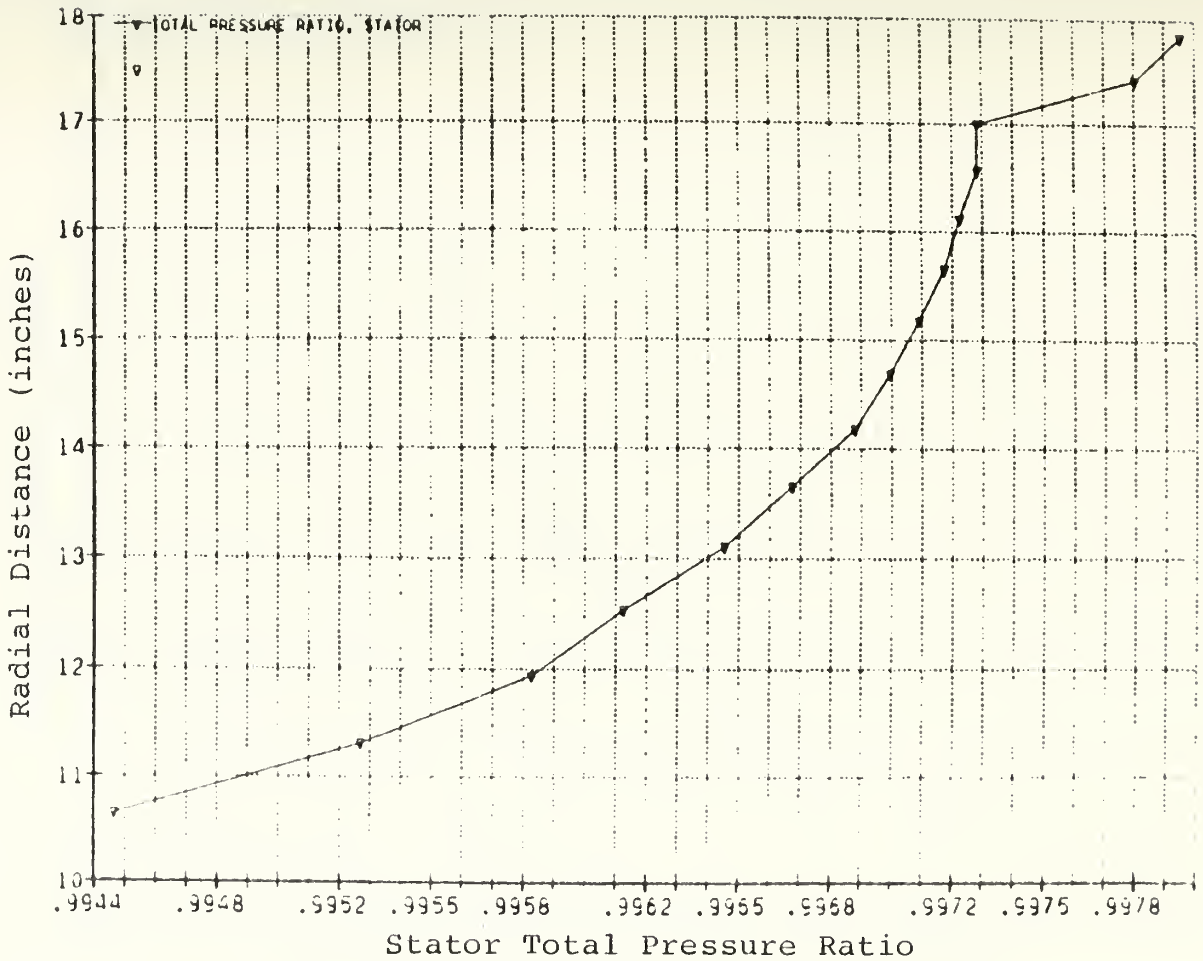

Figure 17. A TURBO Generated Tektonix 618 Plot of the Stator Total Pressure Ratio vs Inlet Radius

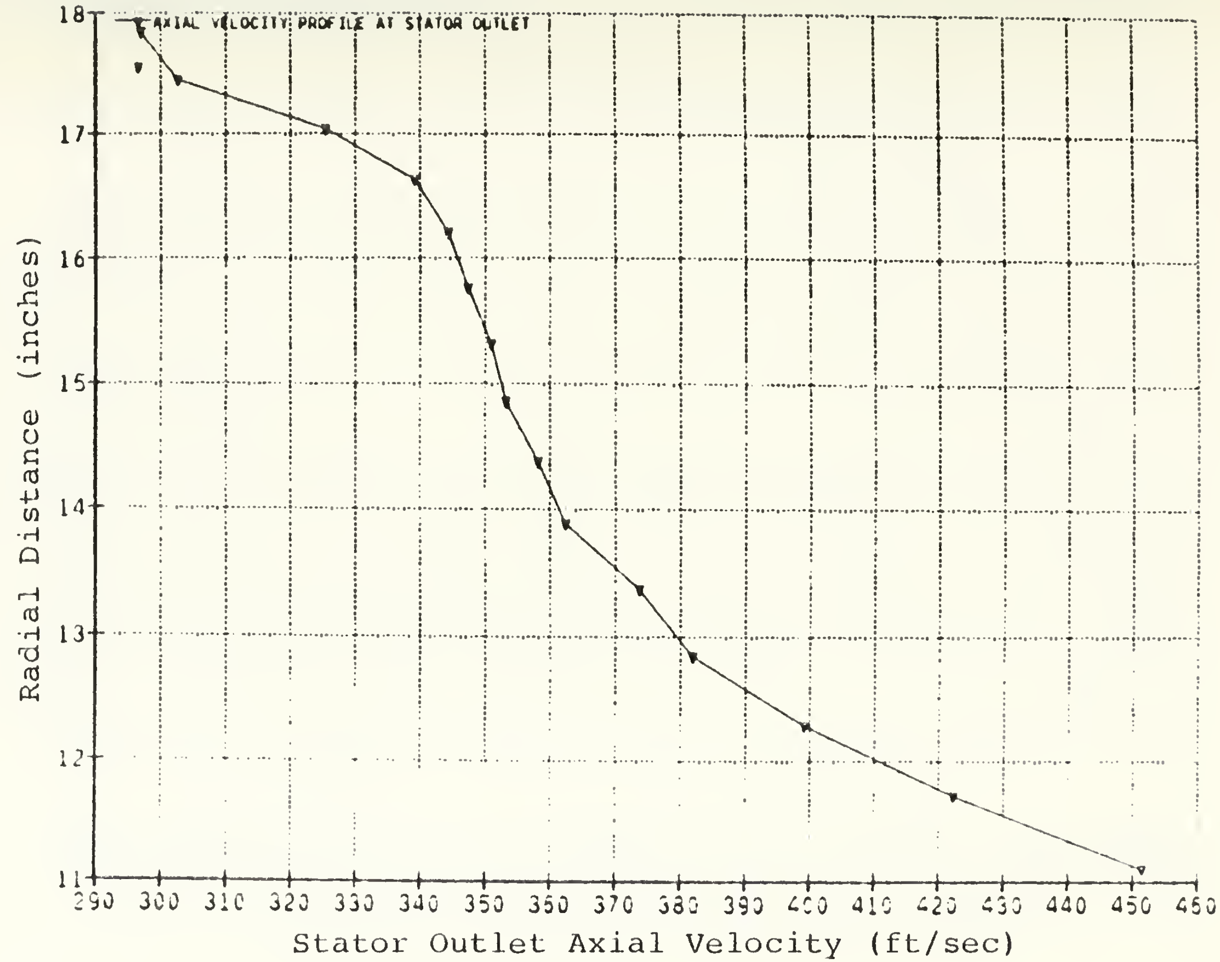

Figure 18. A TURBO Generated Tektonix 618 Plot of the Stator Outlet Axial Velocity

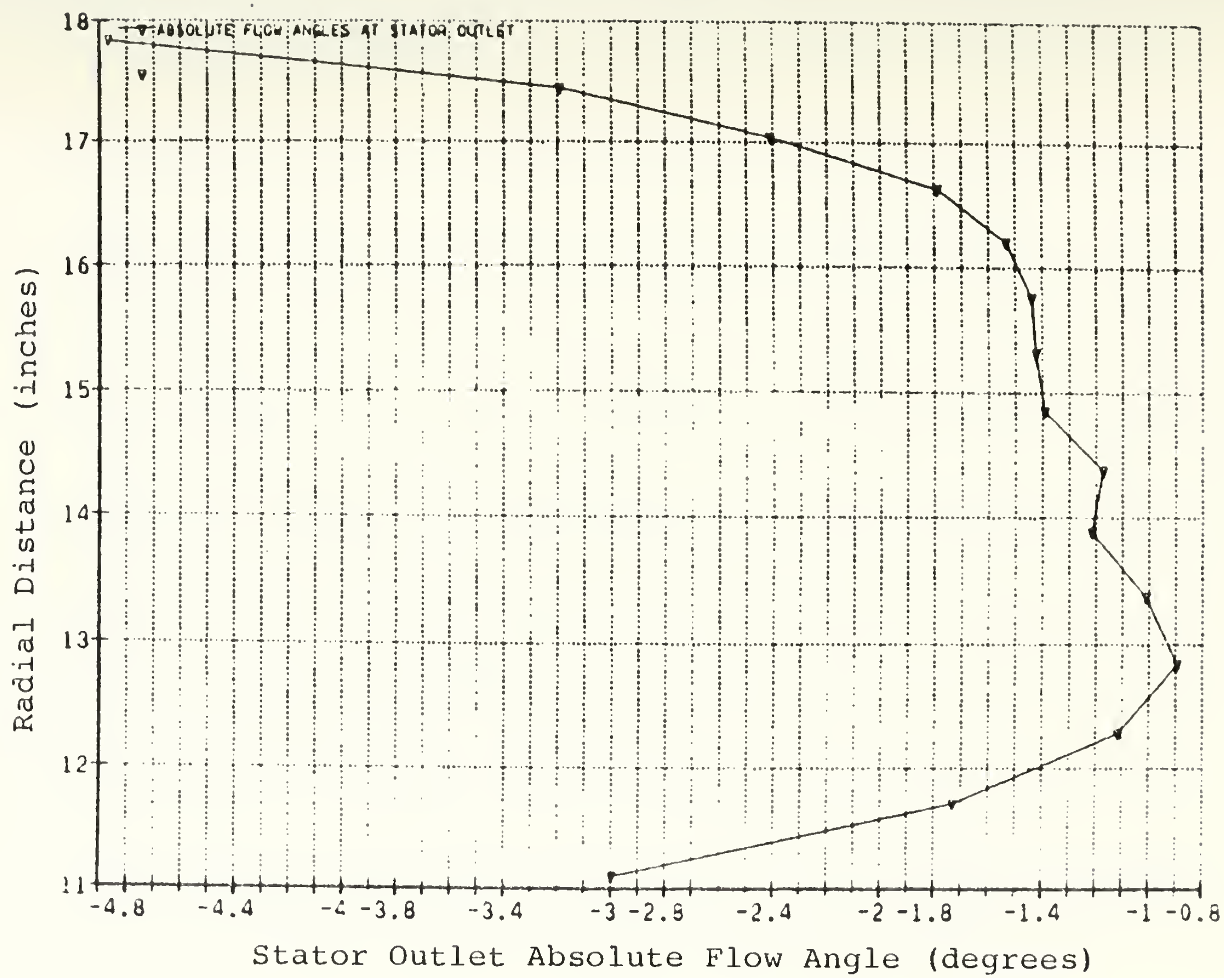

Figure 19. A TURBO Generated Tektonix 618 Plot of the Stator Outlet Absolute Flow Angles

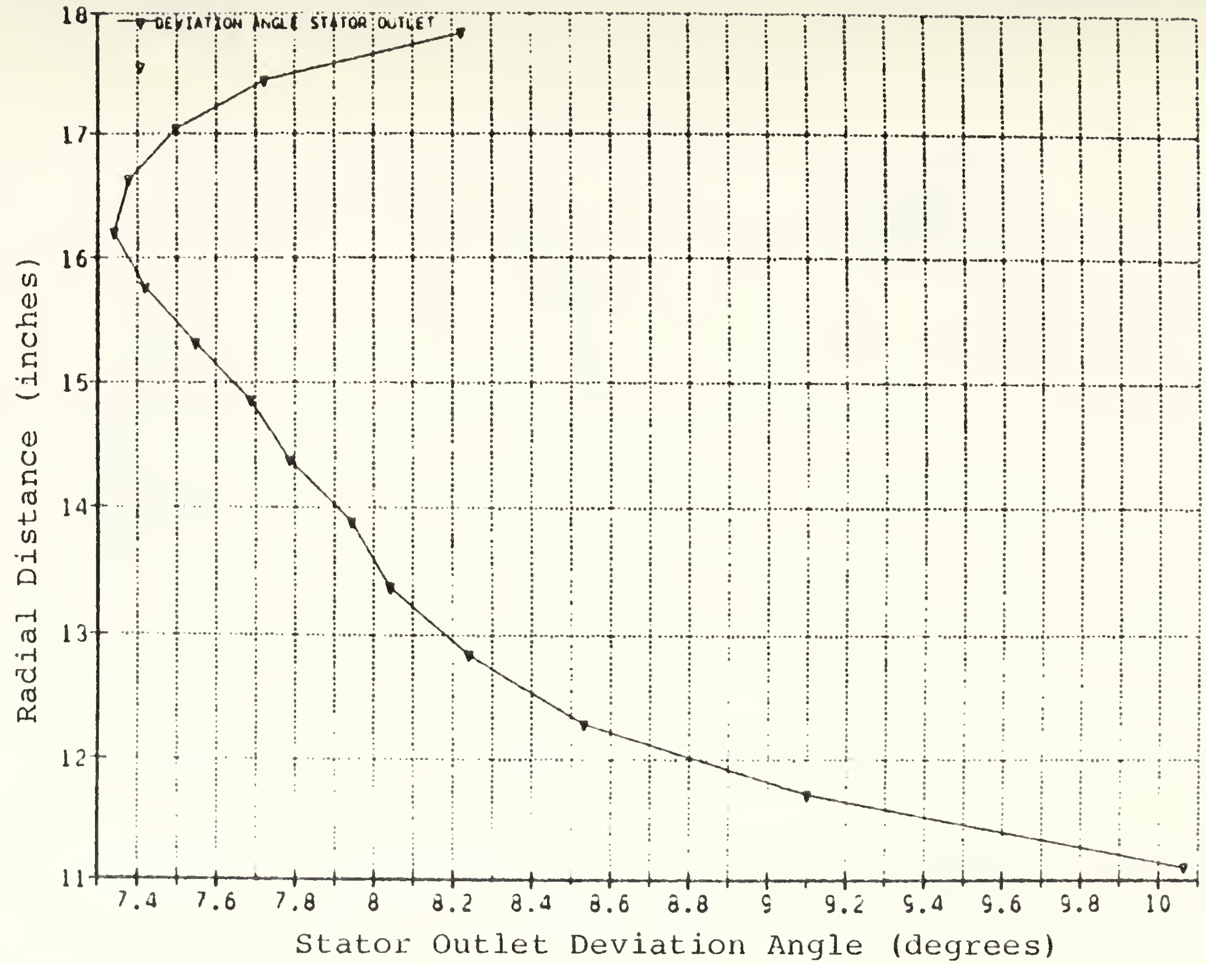

Figure 20. A TURBO Generated Tektonix 618 Plot of the Stator Outlet Deviation Flow Angles

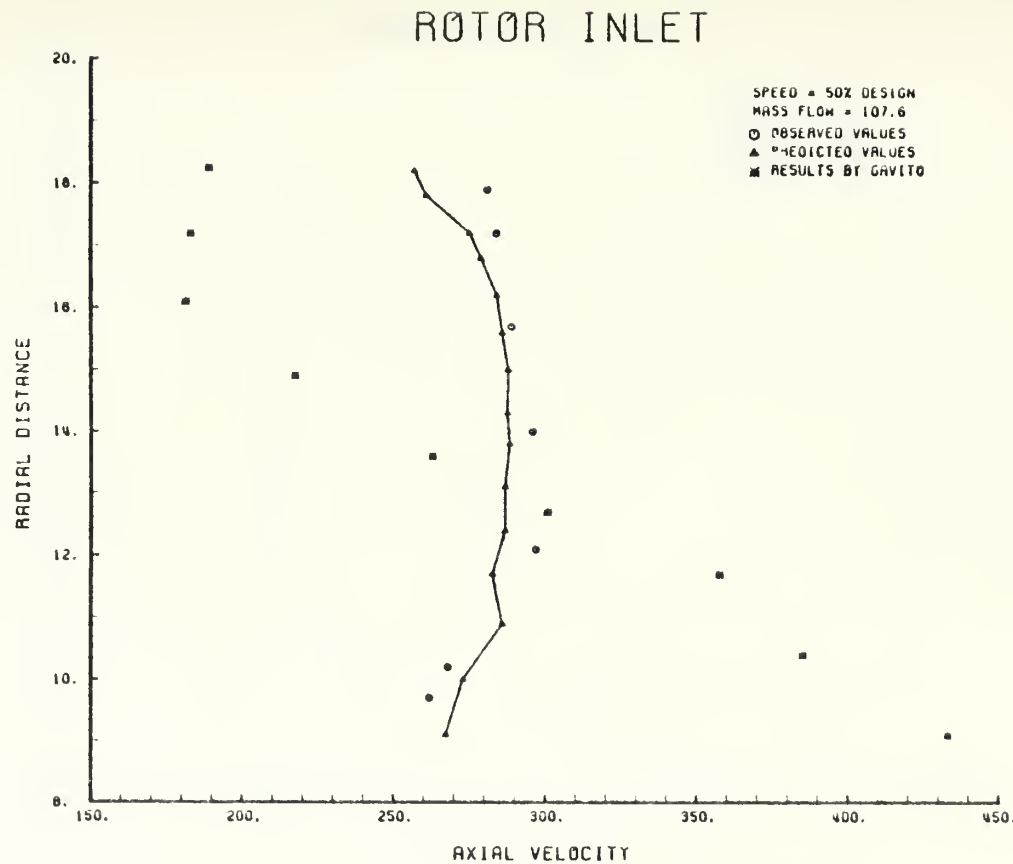

Figure 21. Comparison of Predictions to Gavito and Observations for Rotor Inlet Axial Velocity, 50% Design

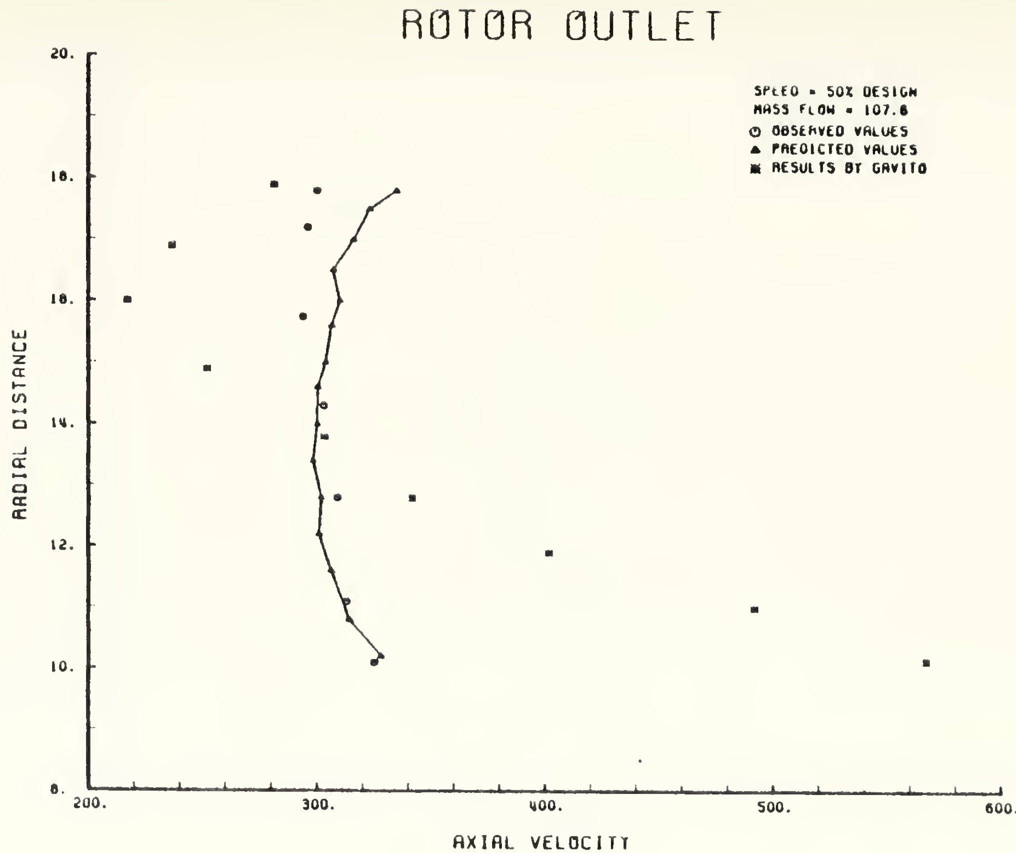

Figure 22. Comparison of Predictions to Gavito and Observations for Rotor Outlet Axial Velocity, 50% Design

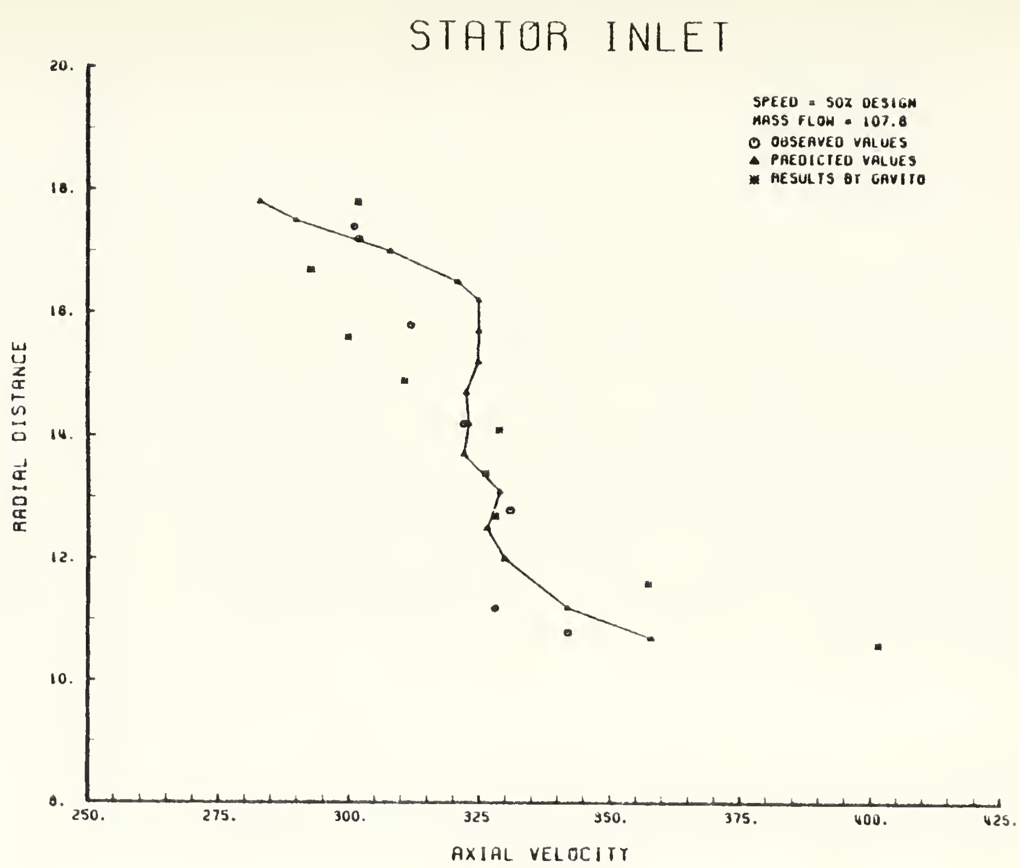

Figure 23. Comparison of Predictions to Gavito and Observations for Stator Inlet Axial Velocity, 50% Design

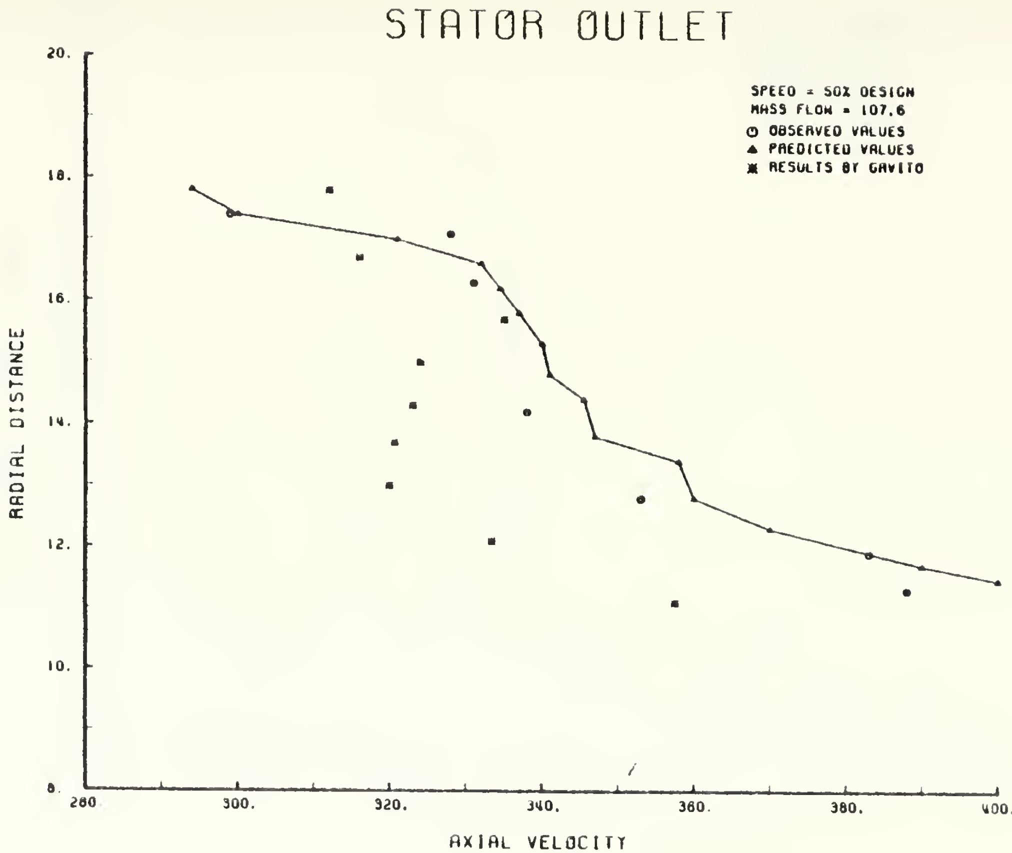

Figure 24. Comparison of Predictions to Gavito and Observations for Stator Outlet Axial Velocity, 50% Design

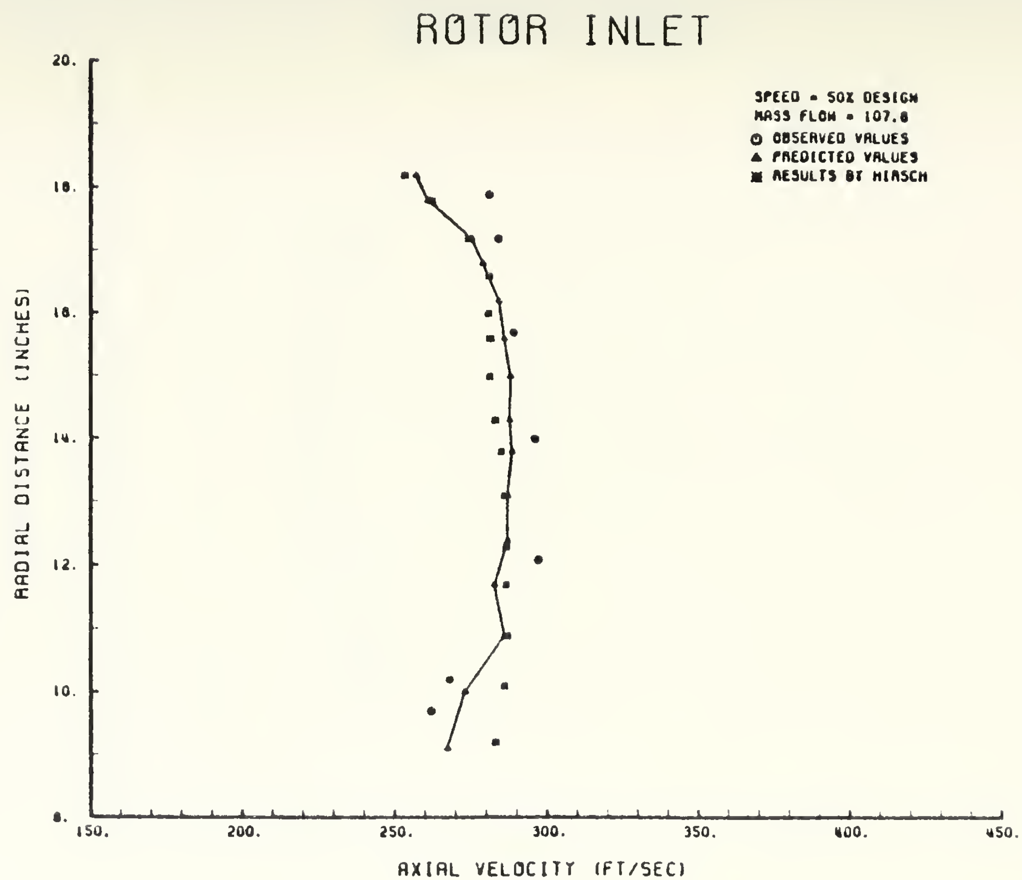

Figure 25. Comparison of Predictions to Hirsch and Observations for Rotor Inlet Axial Velocity, 50% Design

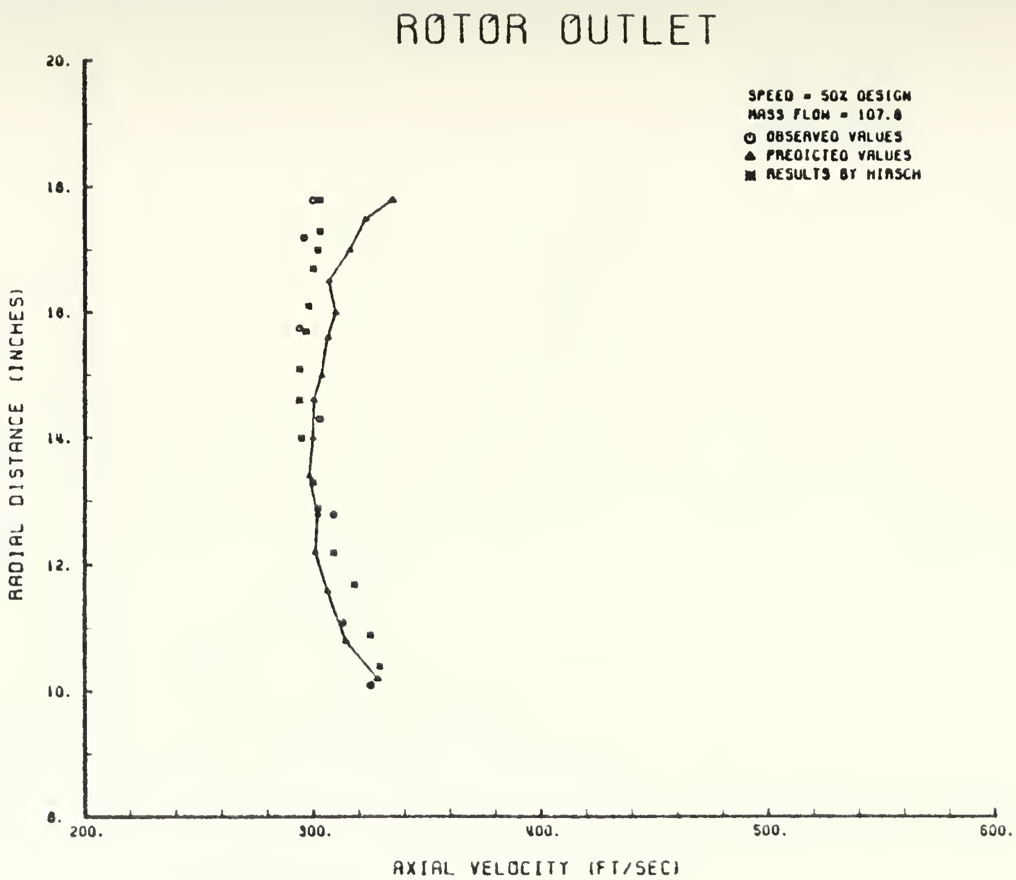

Figure 26. Comparison of Predictions to Hirsch and Observations for Rotor Outlet Axial Velocity, 50% Design

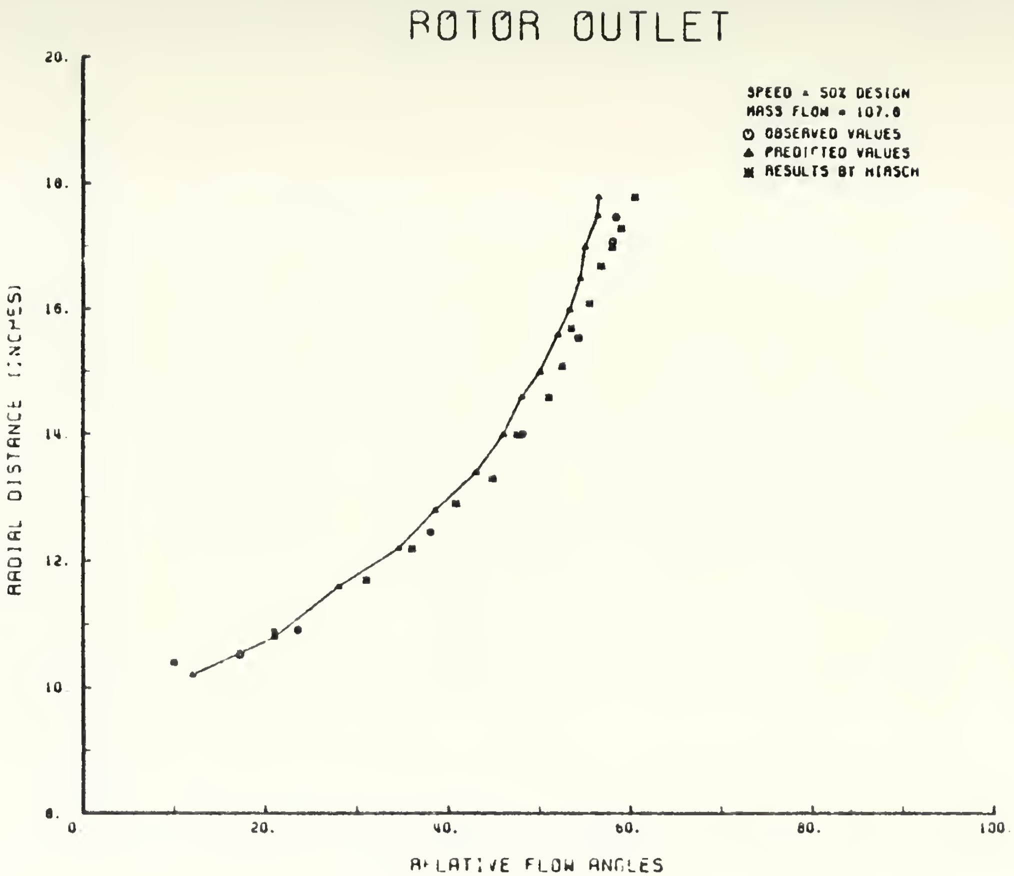

Figure 27. Comparison of Predictions to Hirsch and Observations for Rotor Outlet Relative Flow Angles, 50% Design

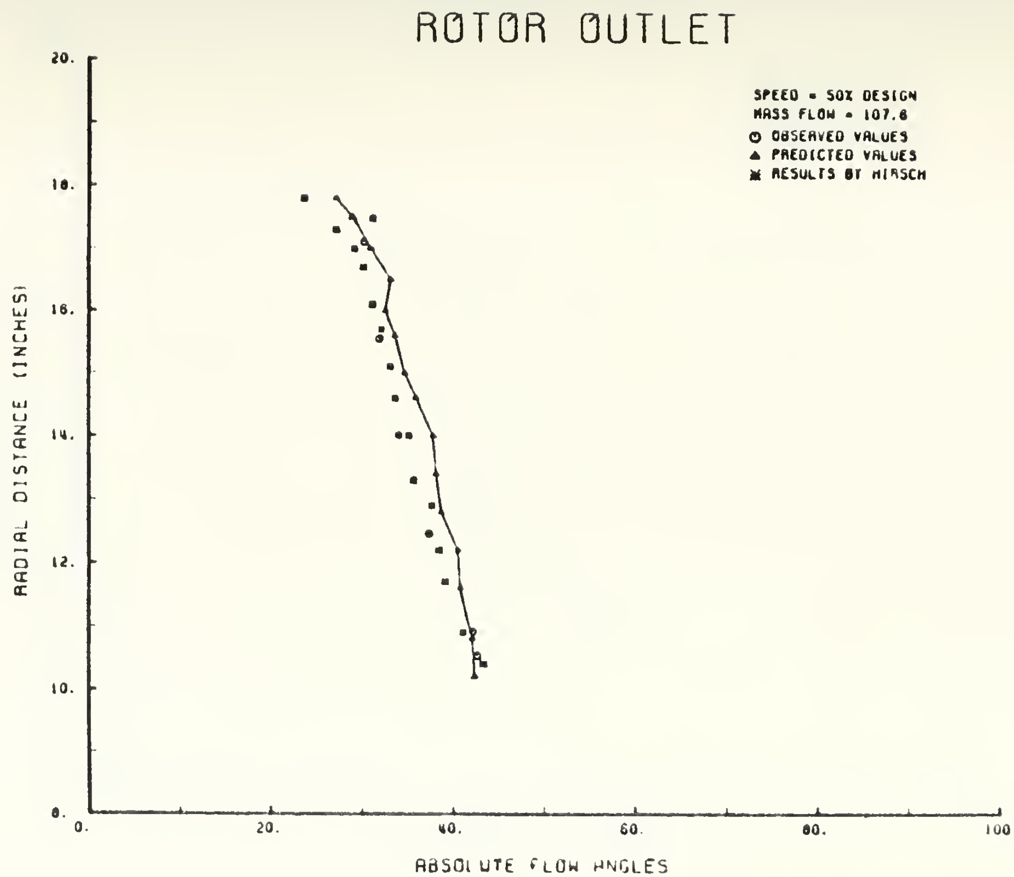

Figure 28. Comparison of Predictions to Hirsch and Observations for Rotor Outlet Absolute Flow Angles, 50% Design

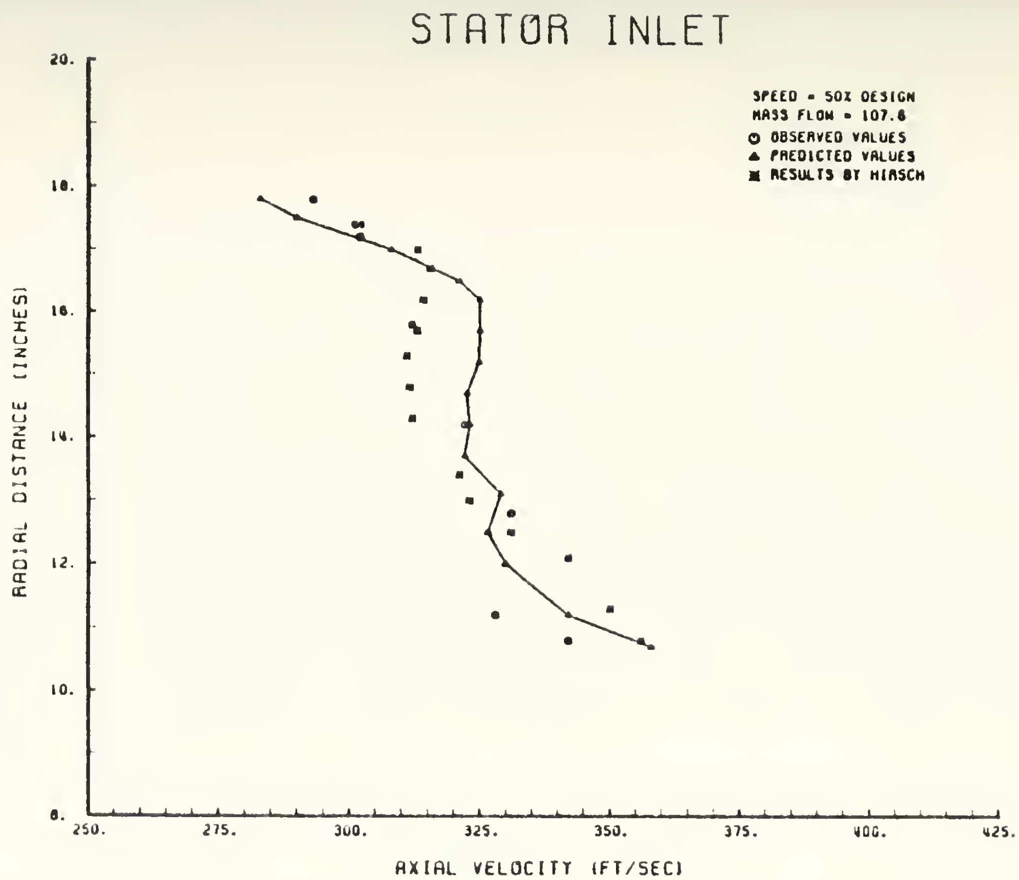

Figure 29. Comparison of Predictions to Hirsch and Observations for Stator Inlet Axial Velocity, 50% Design

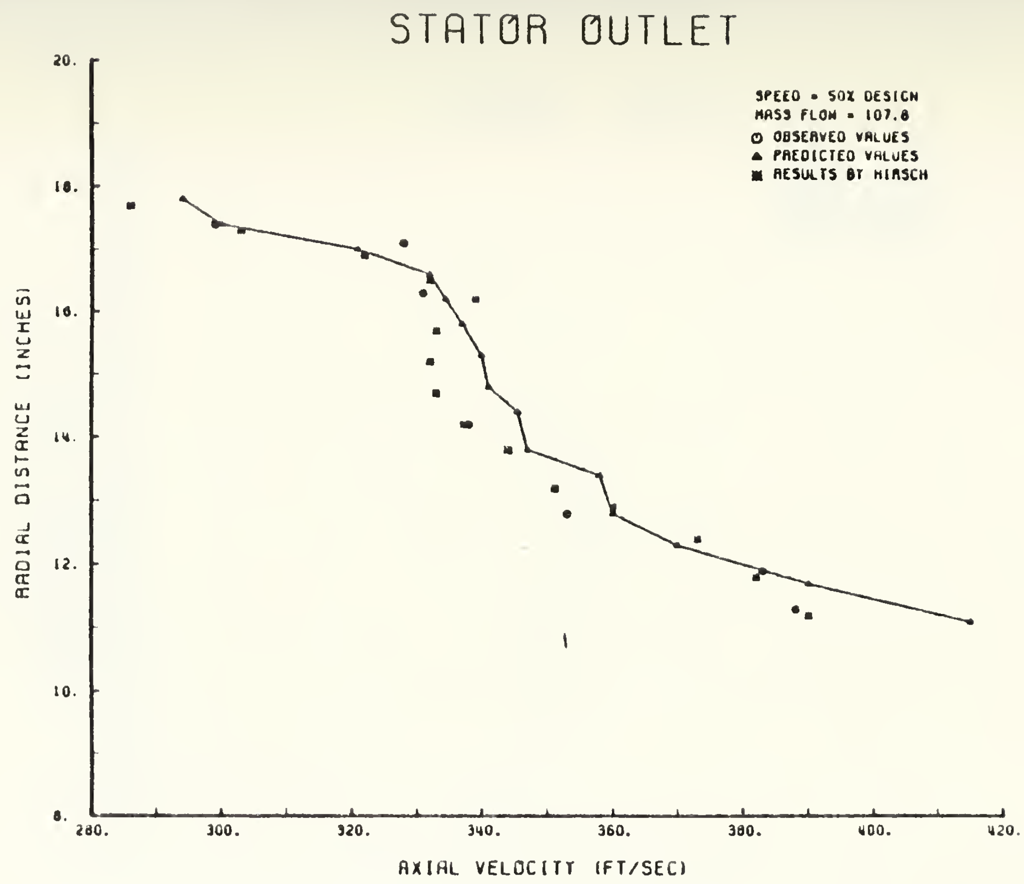

Figure 30. Comparison of Predictions to Hirsch and Observations for Stator Outlet Axial Velocity, 50% Design

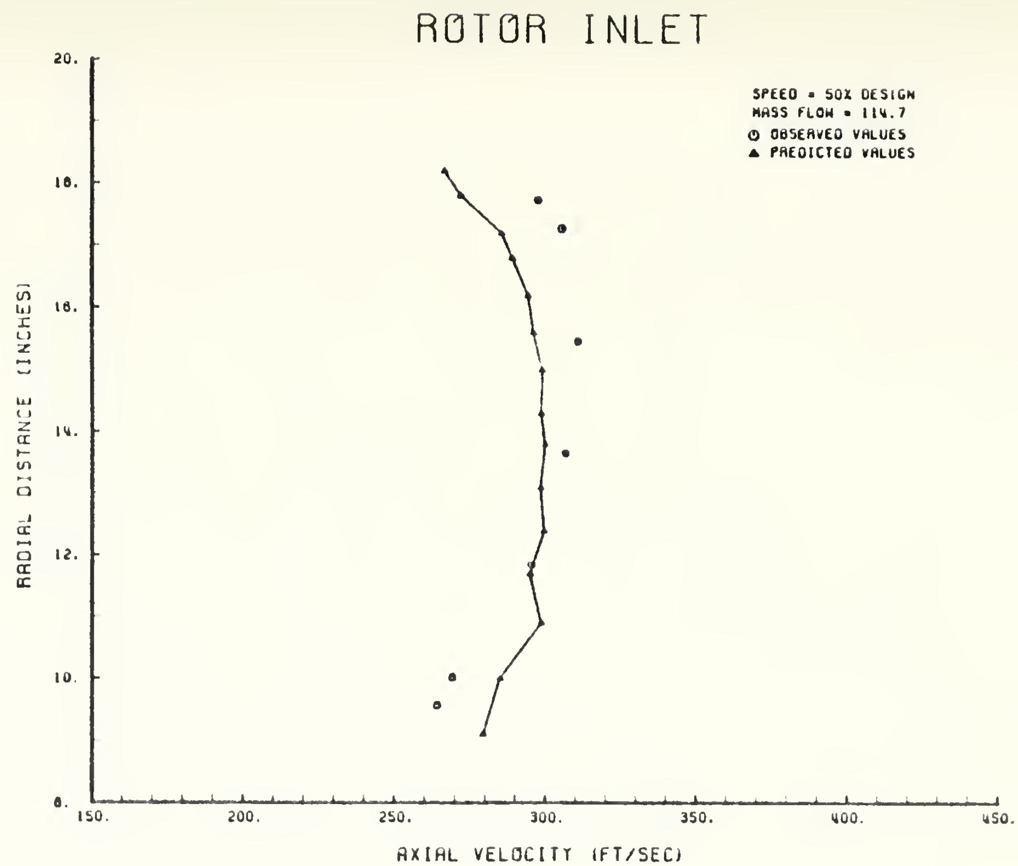

Figure 31. Axial Velocity at the Rotor Inlet, 50% Design

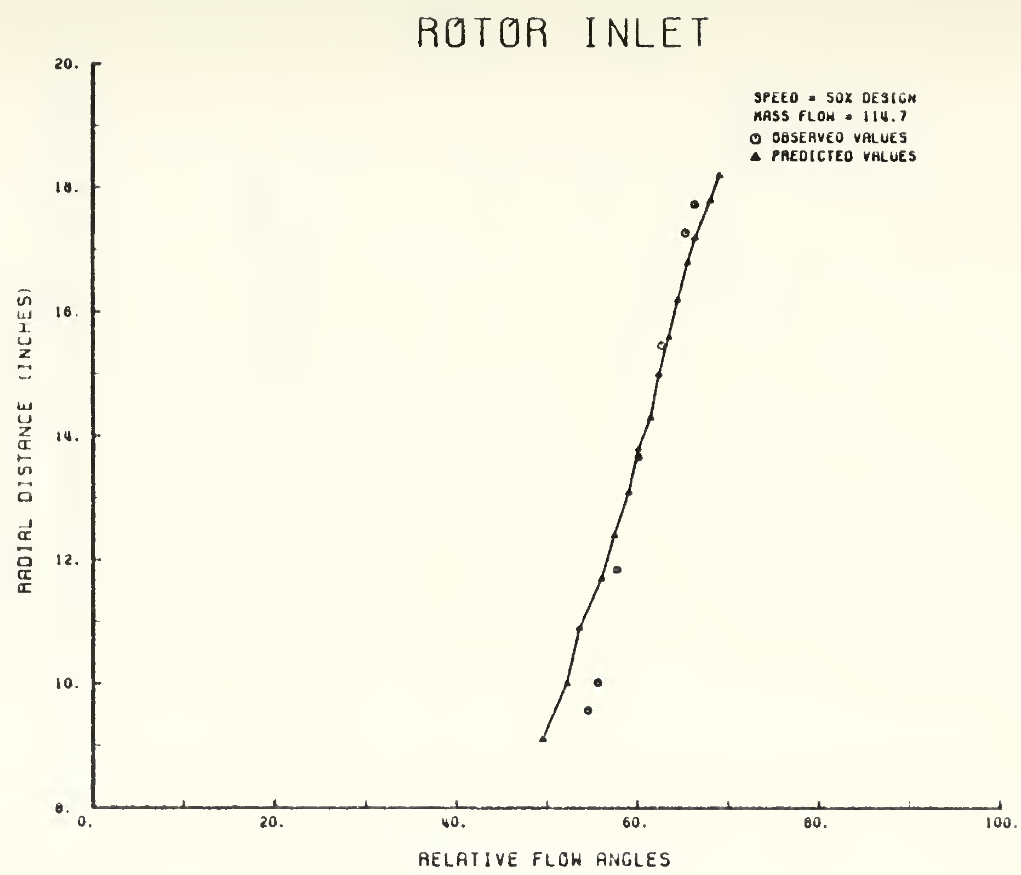

Figure 32. Relative Flow Angles at Rotor Inlet, 50% Design

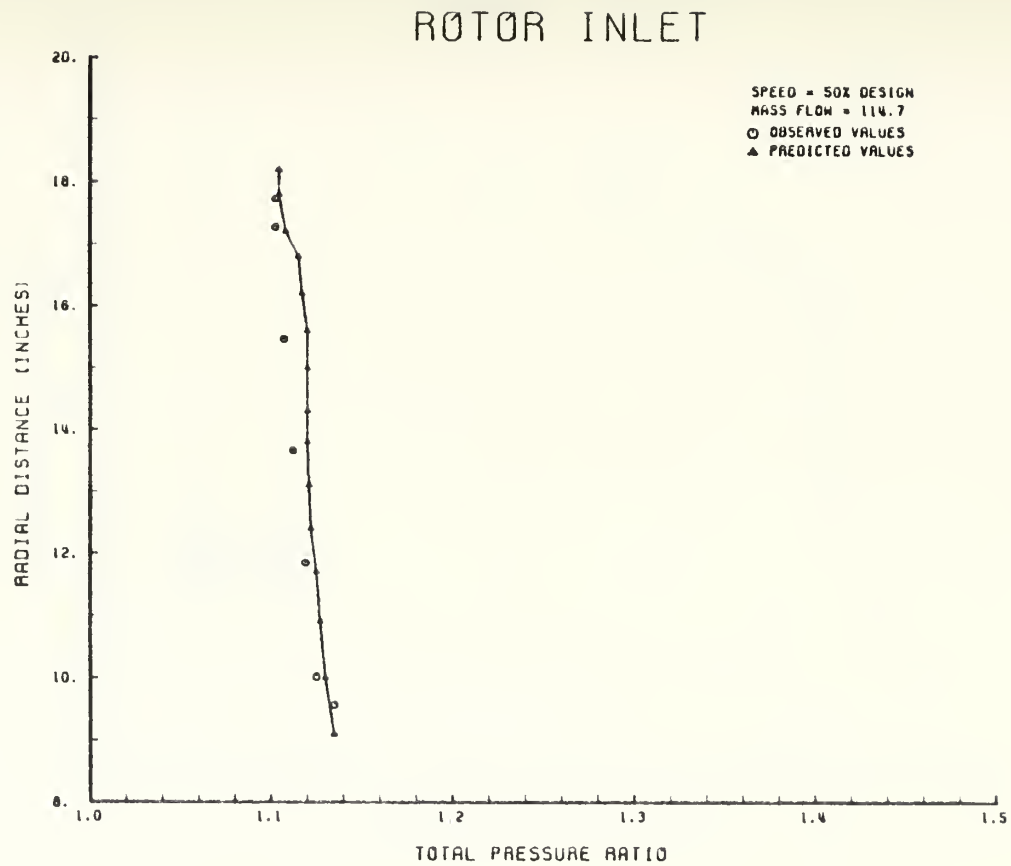

Figure 33. Total Pressure Ratio of the Rotor, 50% Design

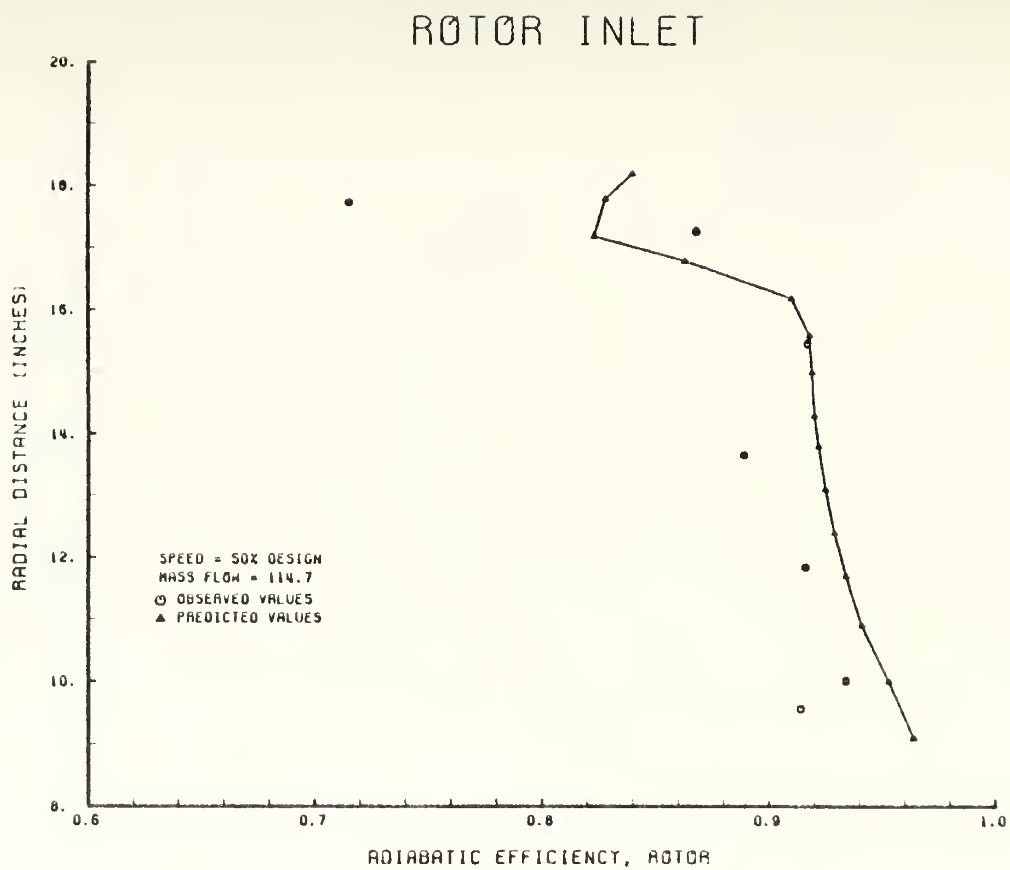

Figure 34. Adiabatic Efficiency of the Rotor, 50% Design

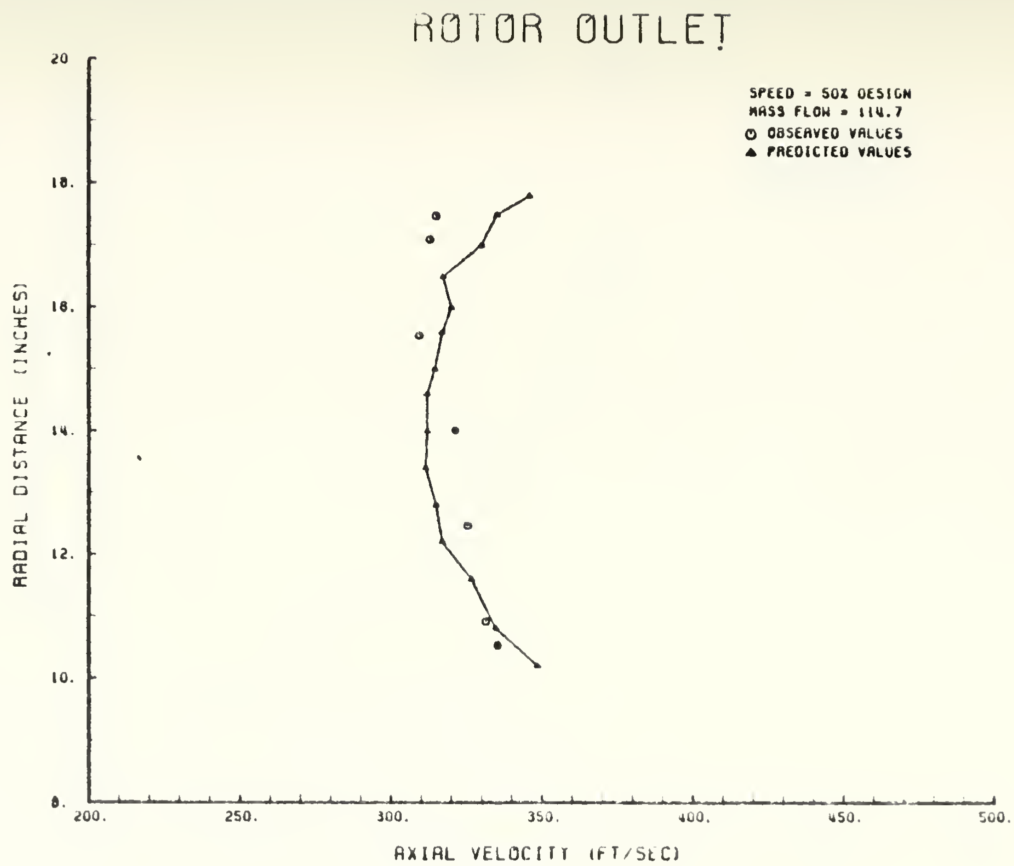

Figure 35. Axial Velocity at the Rotor Outlet, 50% Design

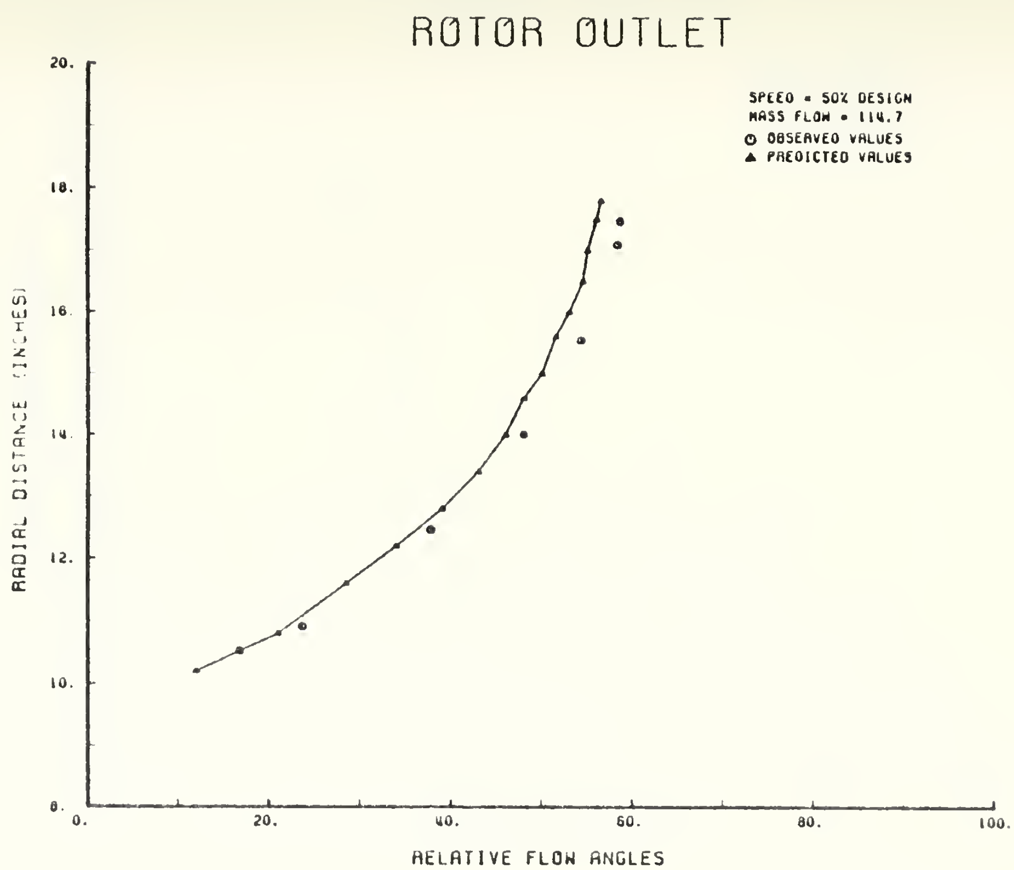

Figure 36. Relative Flow Angles at Rotor Outlet, 50% Design

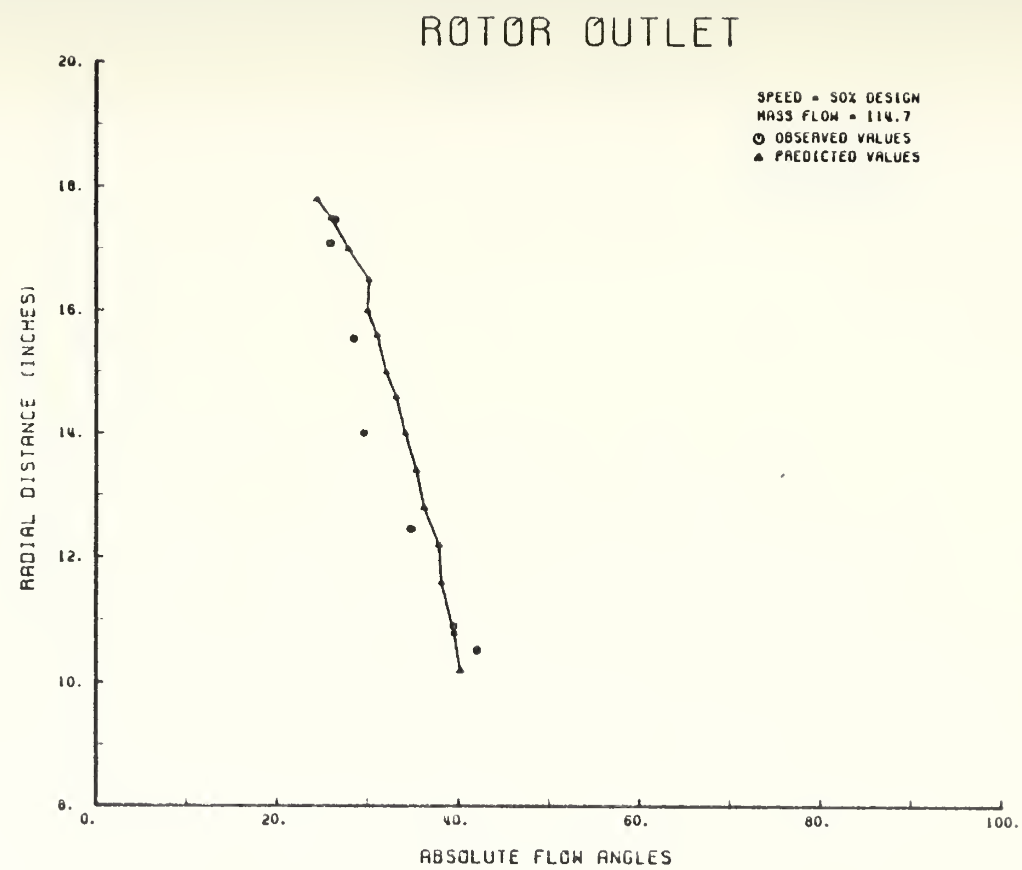

Figure 37. Absolute Flow Angles at Rotor Outlet, 50% Design

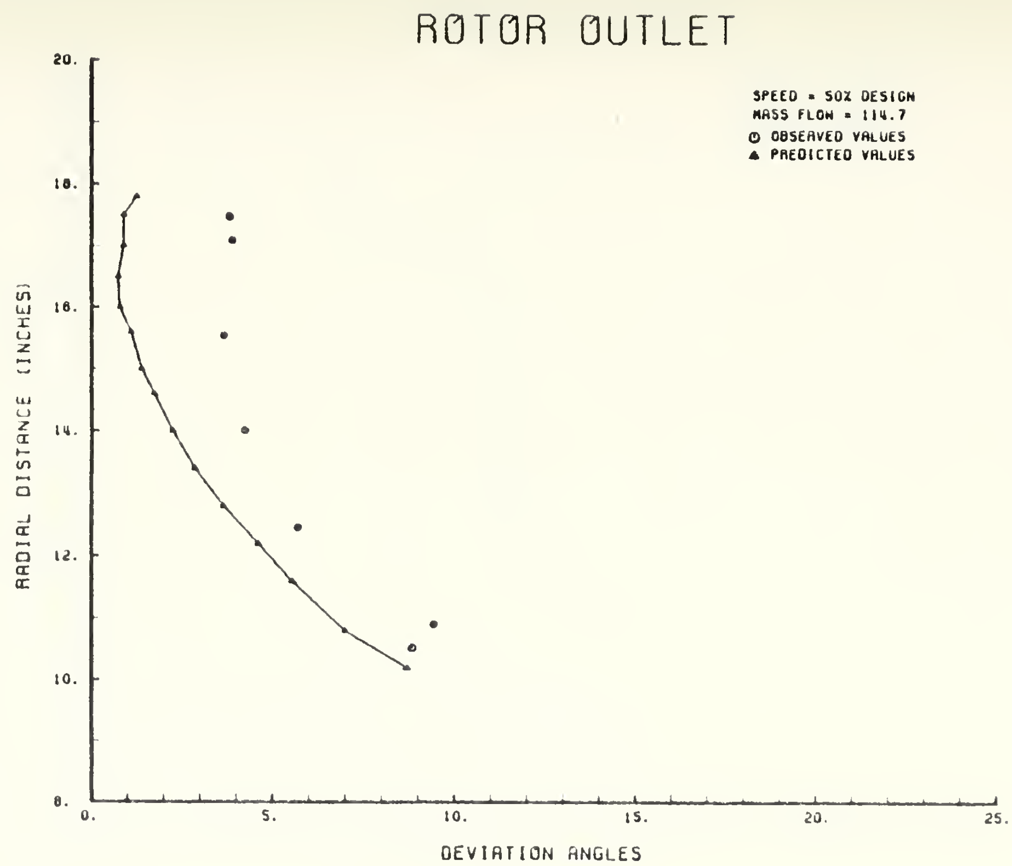

Figure 38. Deviation Flow Angles at Rotor Outlet, 50% Design

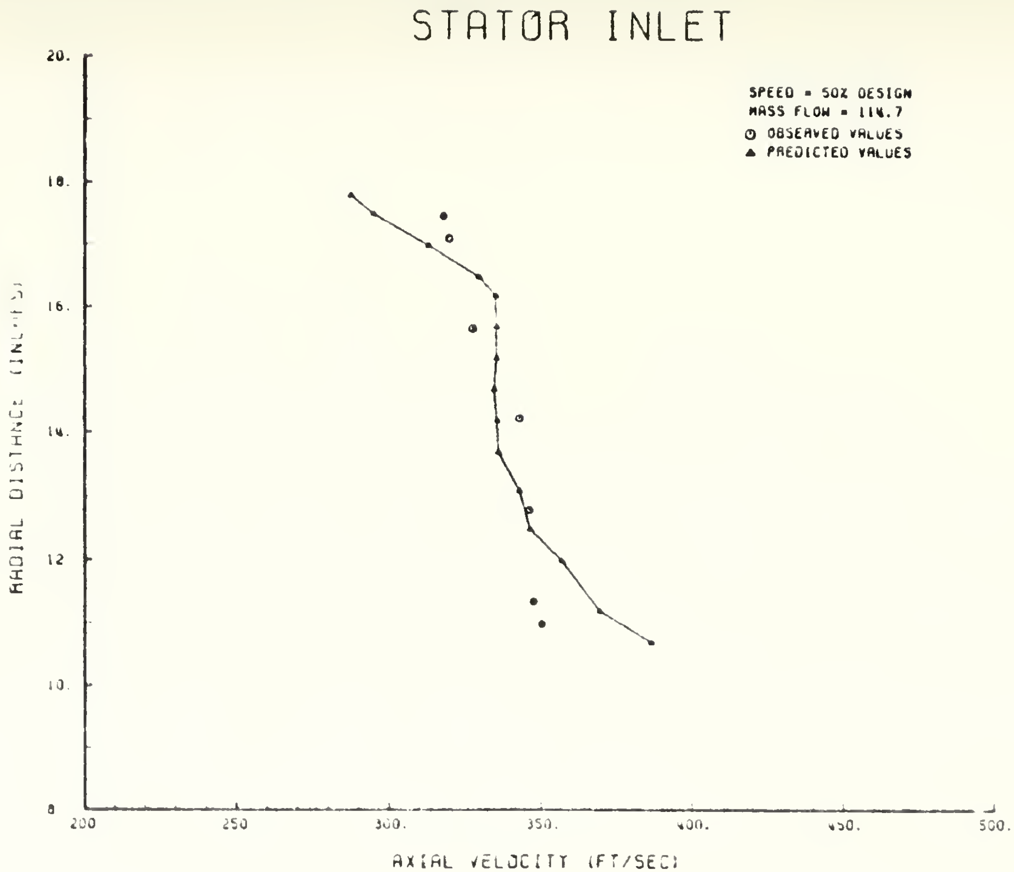

Figure 39. Axial Velocity at the Stator Inlet, 50% Design

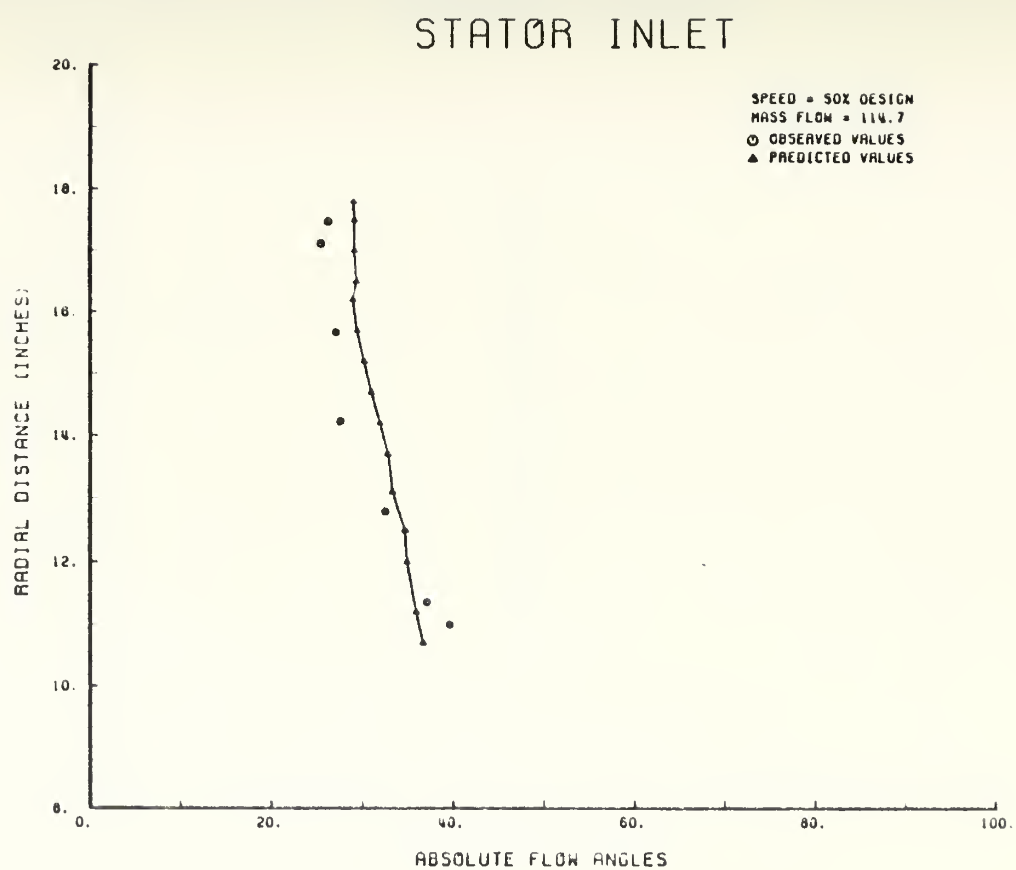

Figure 40. Absolute Flow Angles at Stator Inlet, 50% Design

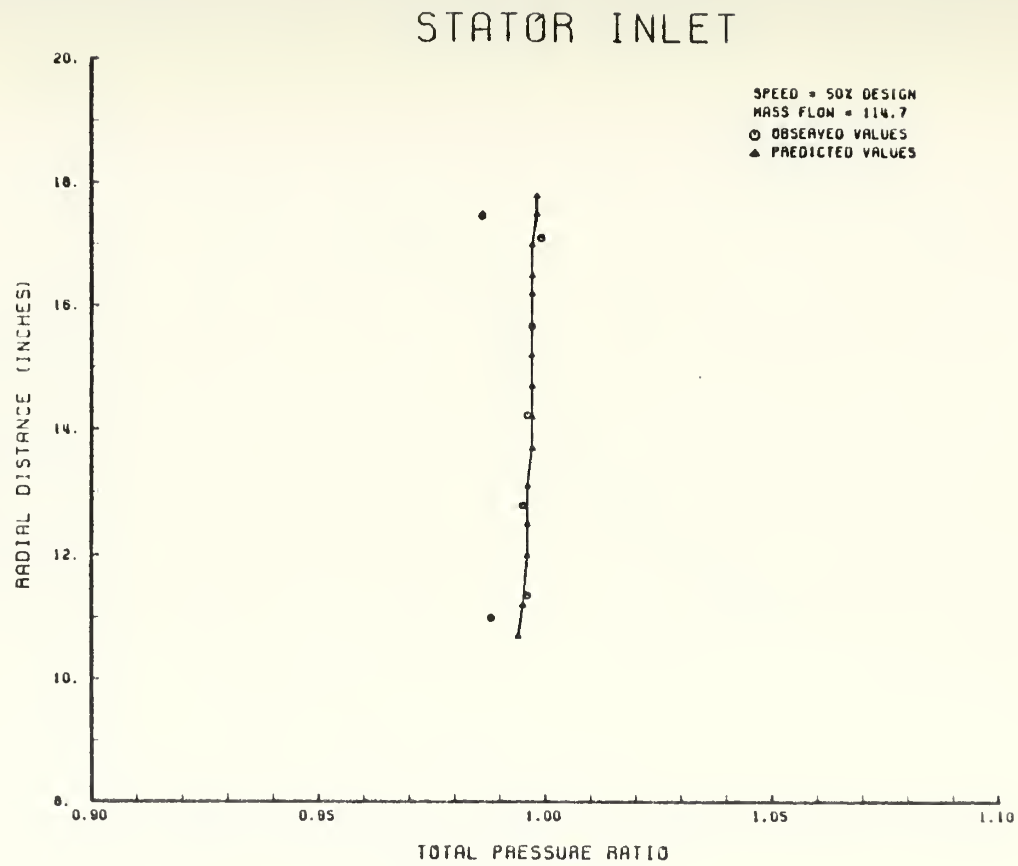

Figure 41. Total Pressure Ratio at Stator Inlet, 50% Design

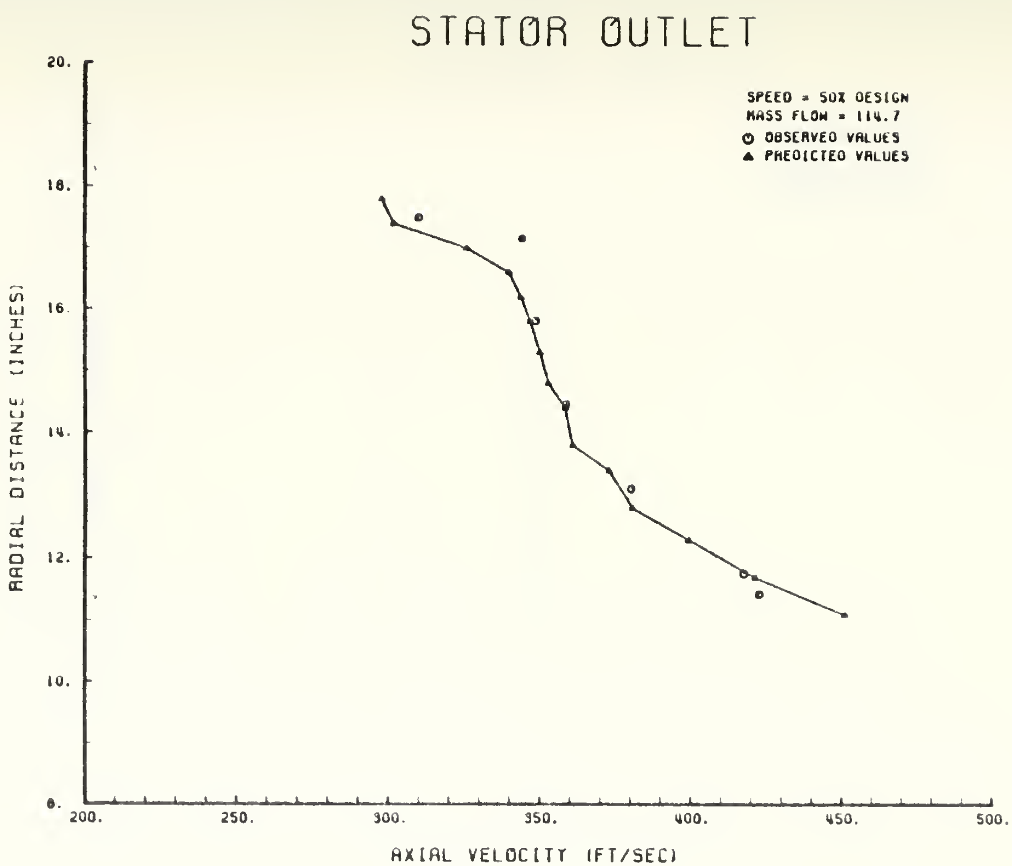

Figure 42. Axial Velocity at the Stator Outlet, 50% Design

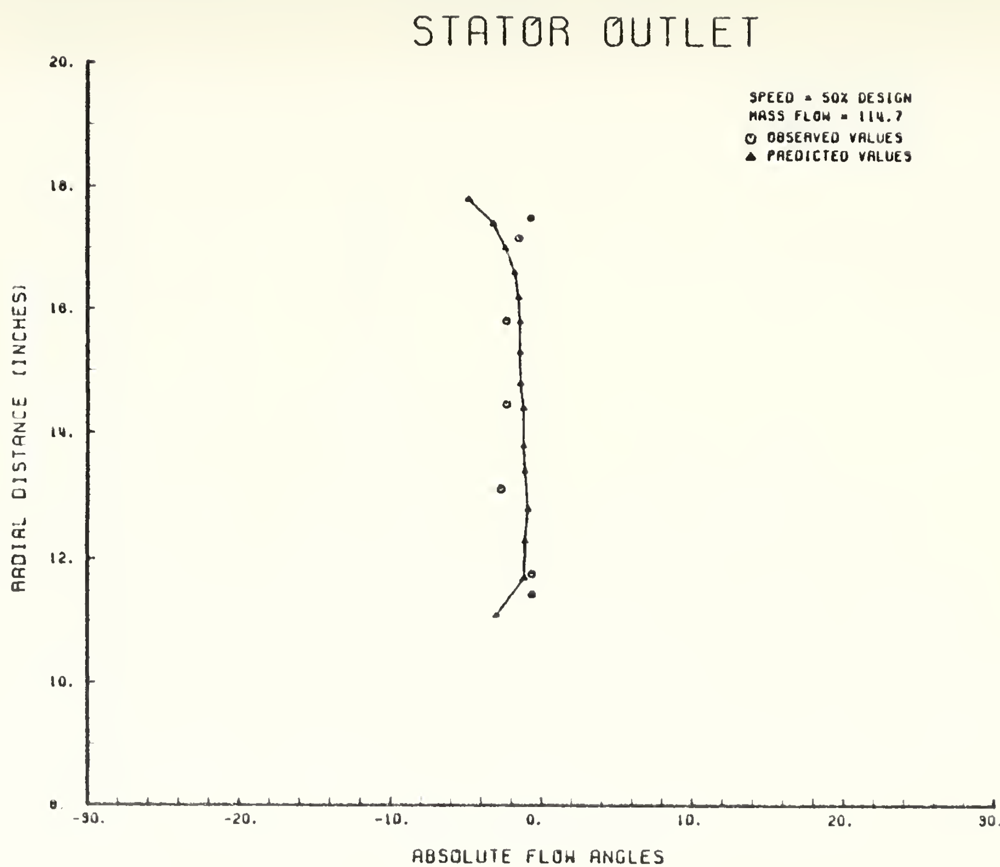

Figure 43. Absolute Flow Angles at Stator Outlet, 50% Design

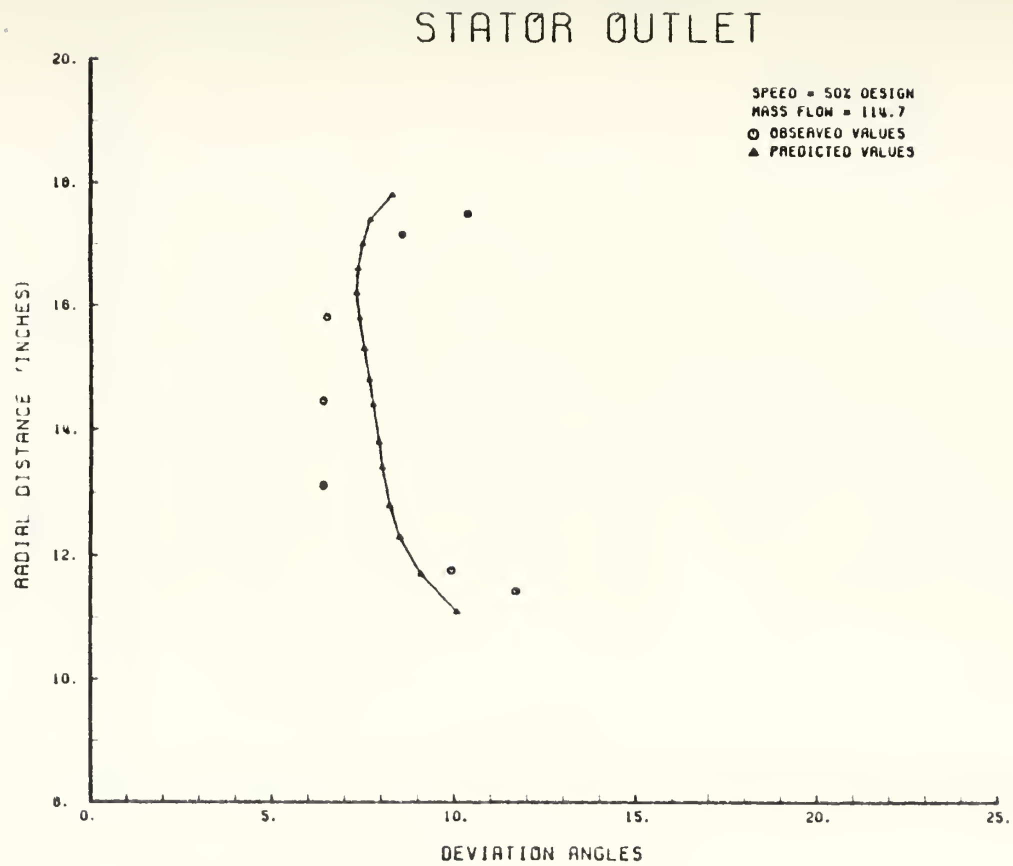

Figure 44. Deviation Flow Angles at Stator Outlet, 50% Design

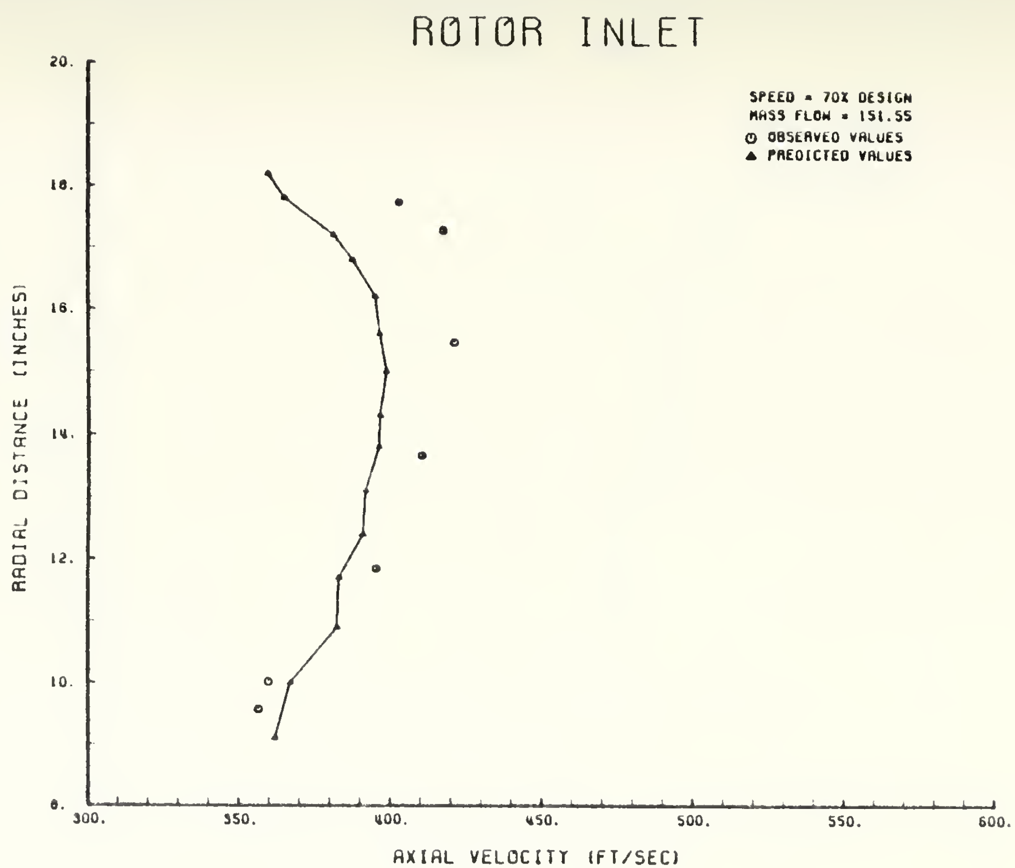

Figure 45. Axial Velocity at the Rotor Inlet, 70% Design

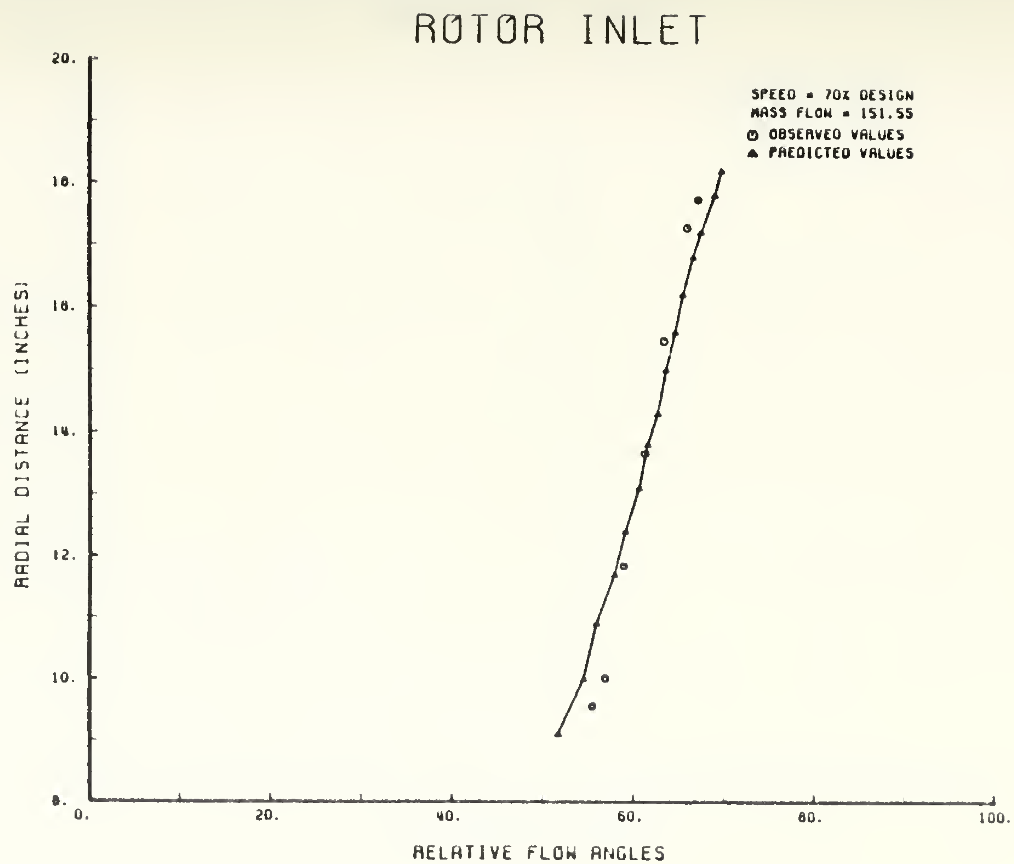

Figure 46. Relative Flow Angles at Rotor Inlet, 70% Design

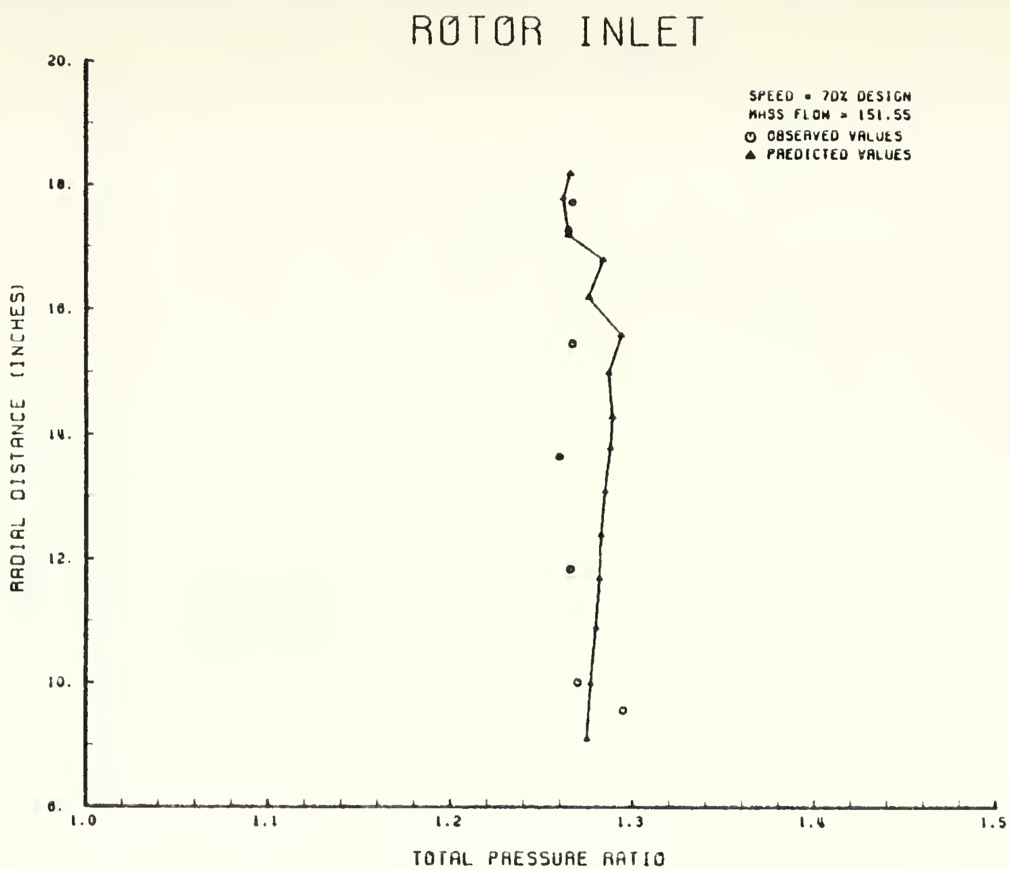

Figure 47. Total Pressure Ratio of the Rotor, 70% Design

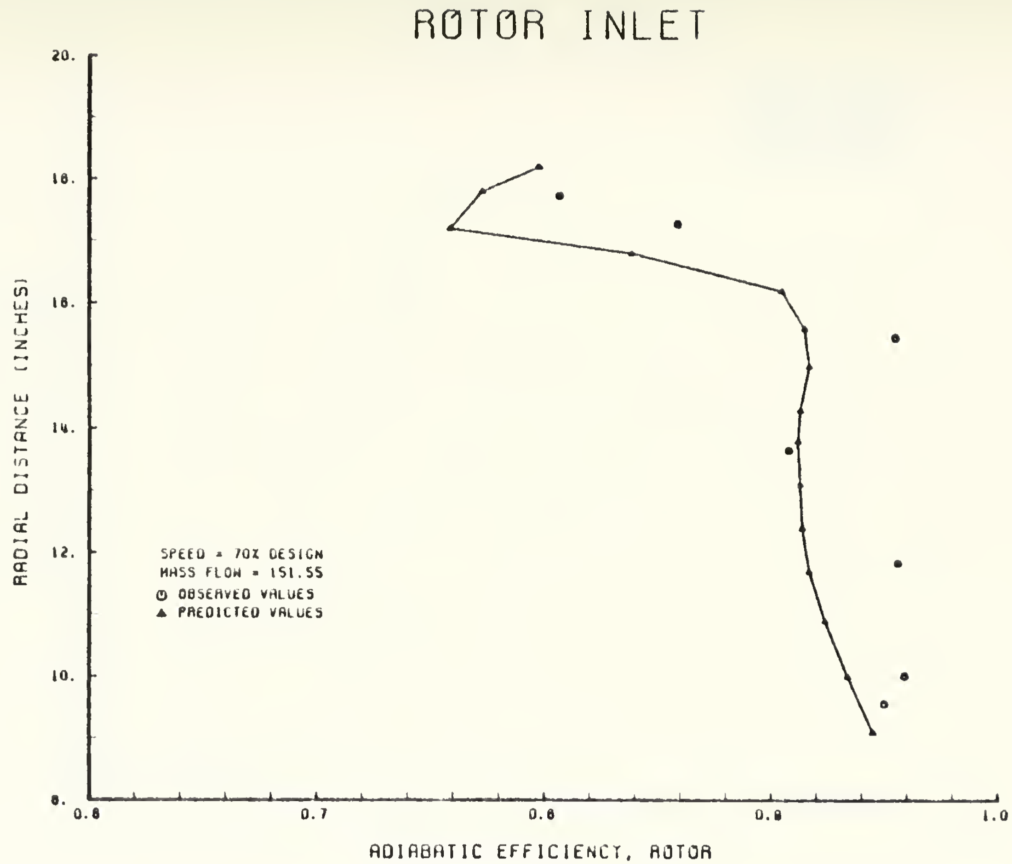

Figure 48. Adiabatic Efficiency of the Rotor, 70% Design

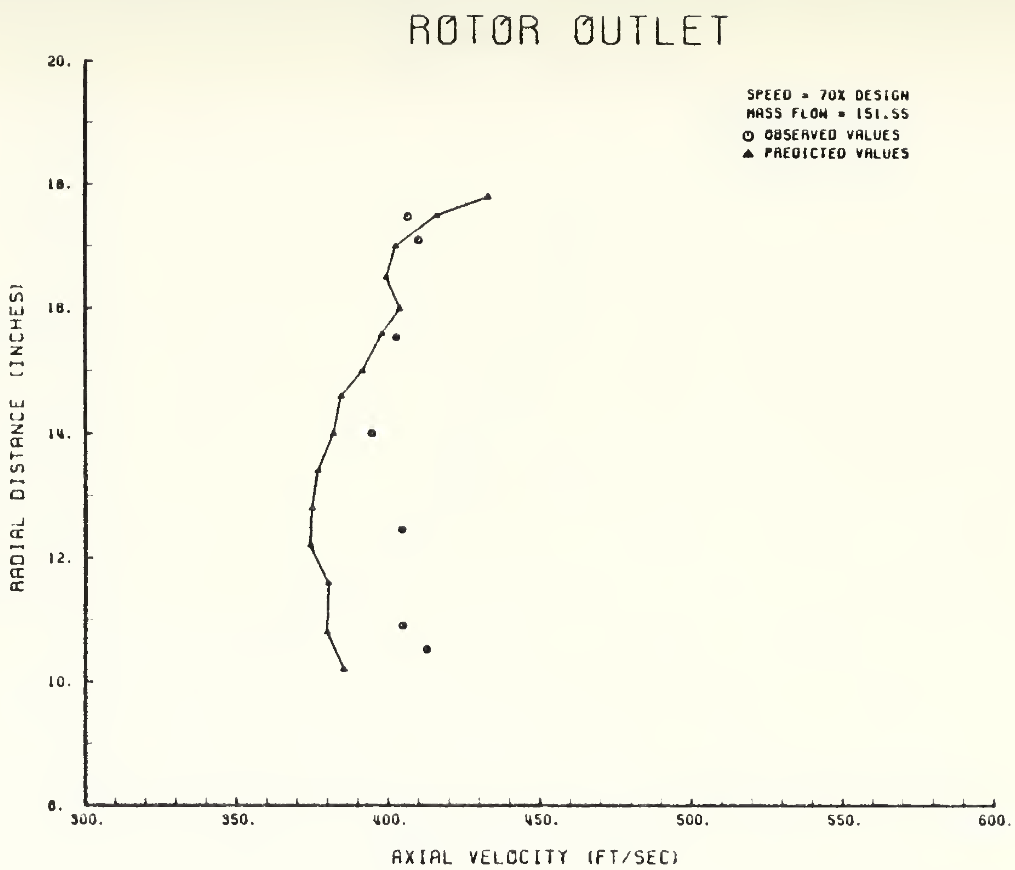

Figure 49. Axial Velocity at the Rotor Outlet, 70% Design

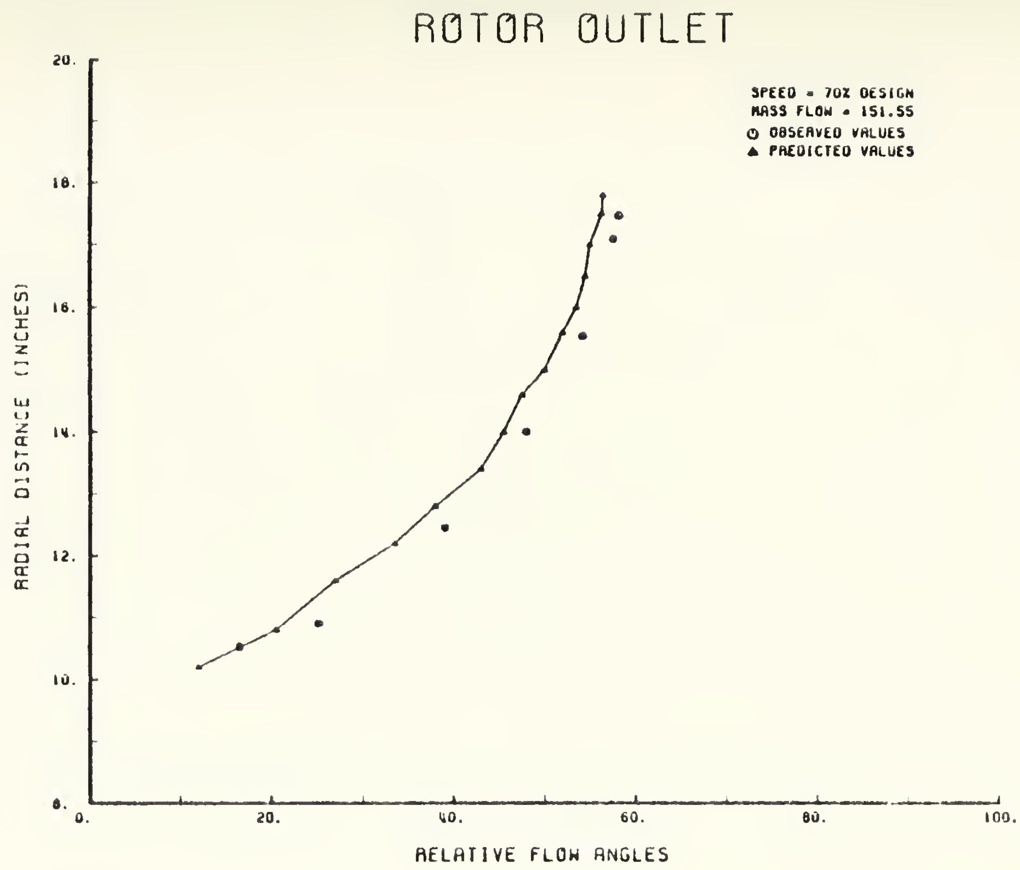

Figure 50. Relative Flow Angles at Rotor Outlet, 70% Design

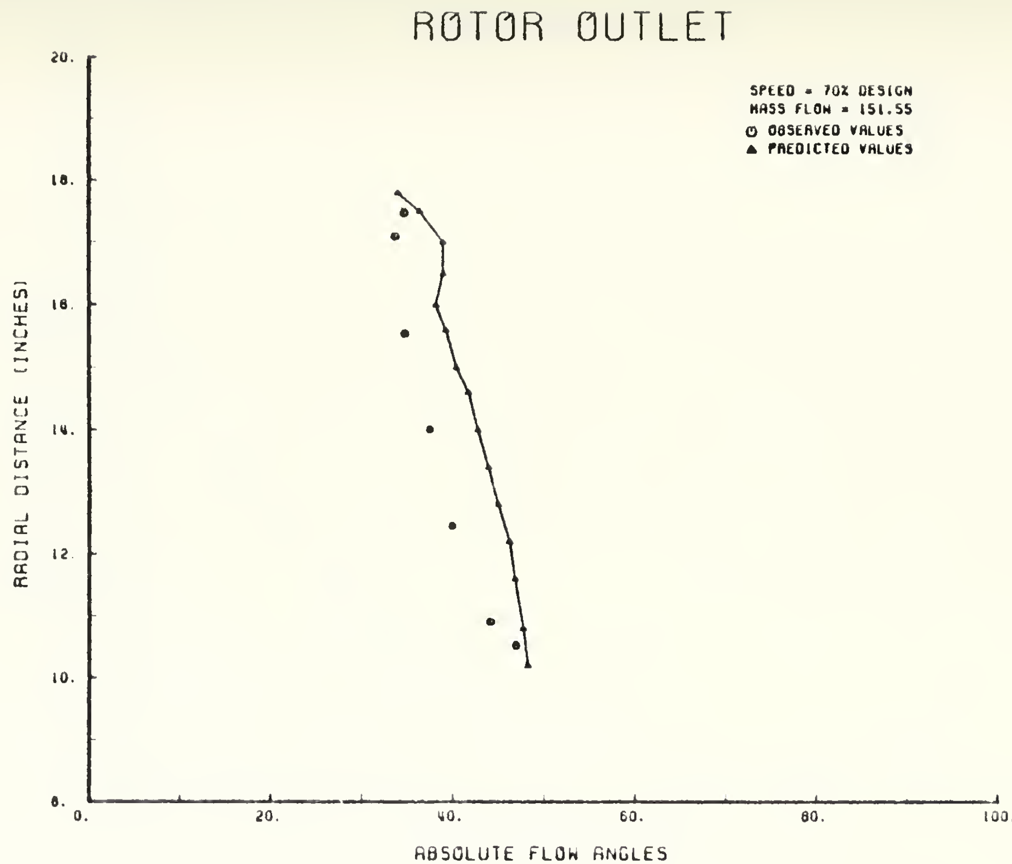

Figure 51. Absolute Flow Angles at Rotor Outlet, 70% Design

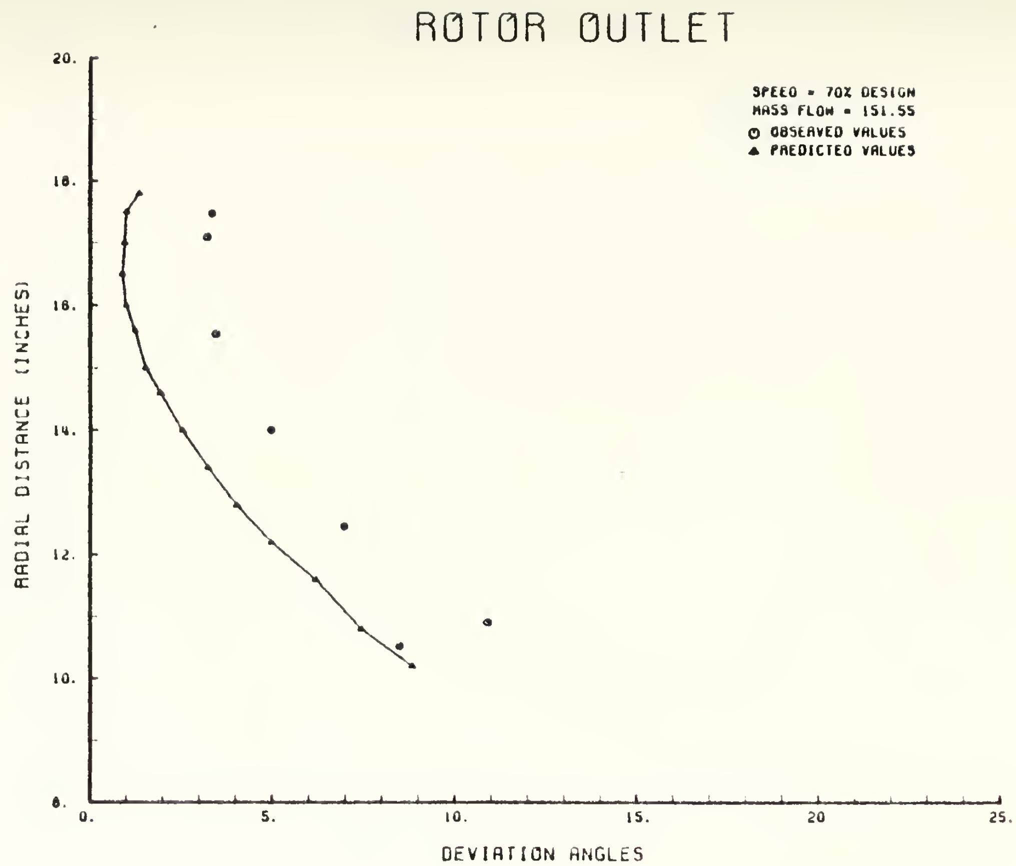

Figure 52. Deviation Flow Angles at Rotor Outlet, 70% Design

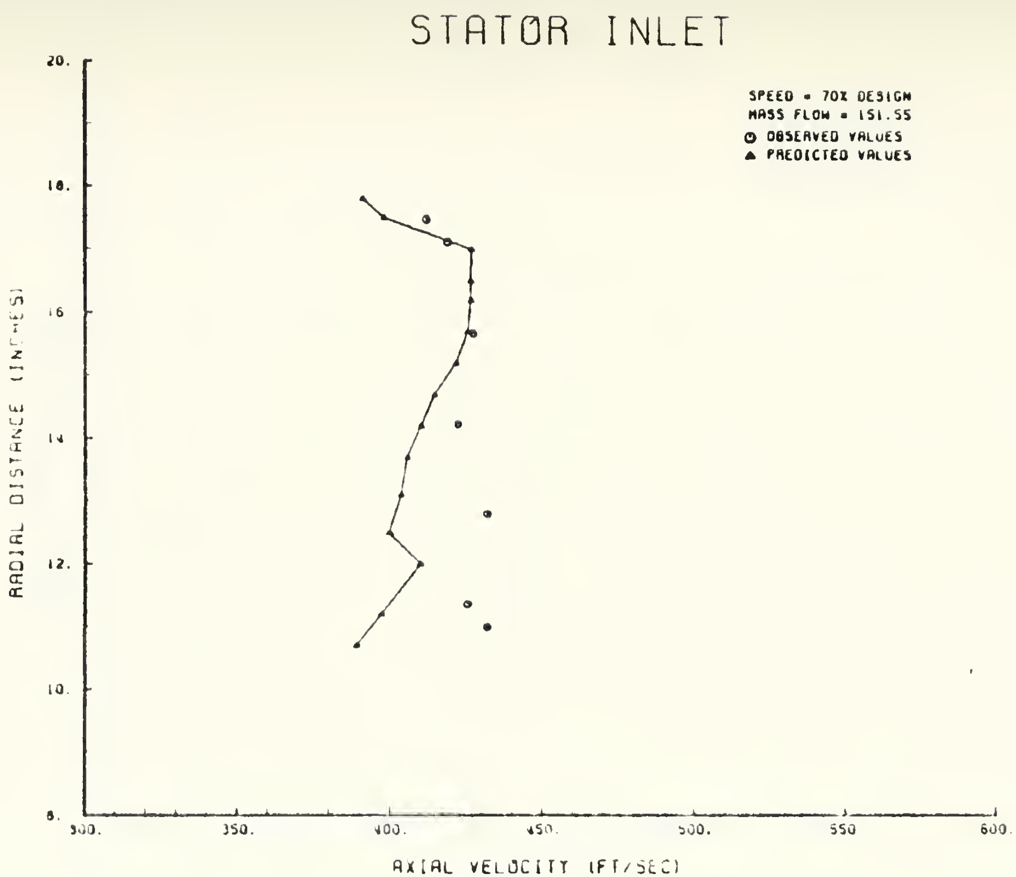

Figure 53. Axial Velocity at the Stator Inlet, 70% Design

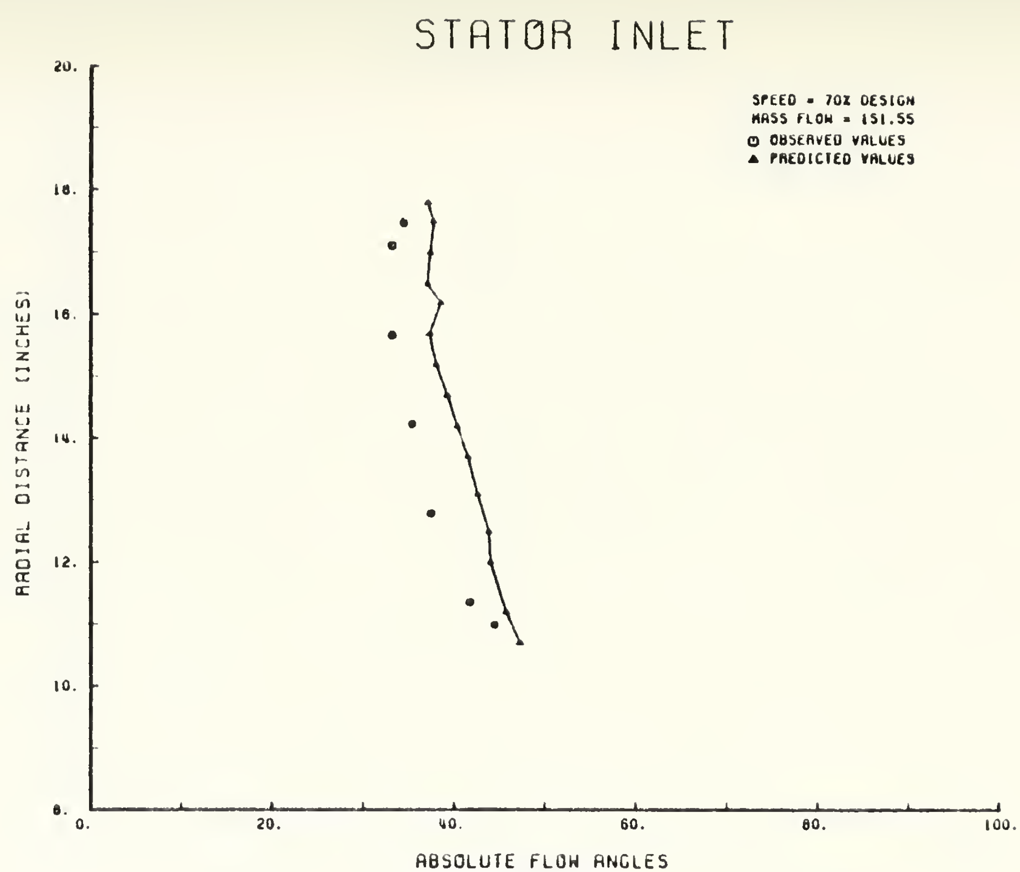

Figure 54. Absolute Flow Angles at Stator Inlet, 70% Design

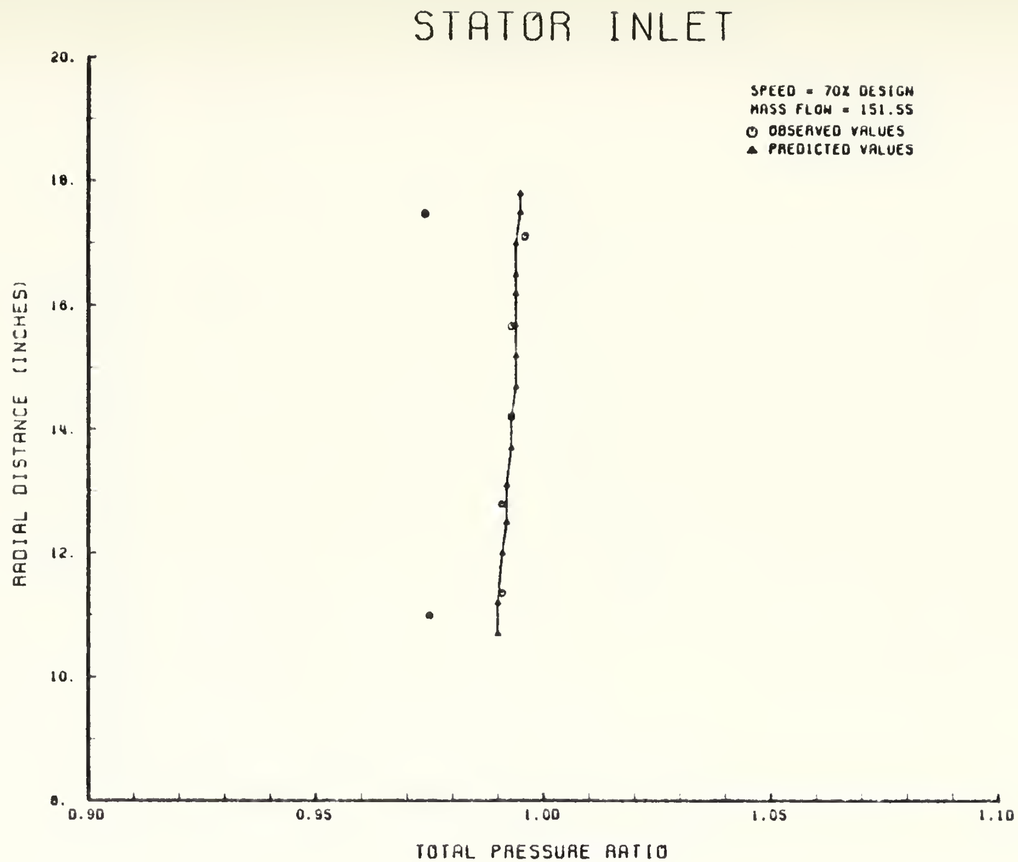

Figure 55. Total Pressure Ratio at Stator Inlet, 70% Design

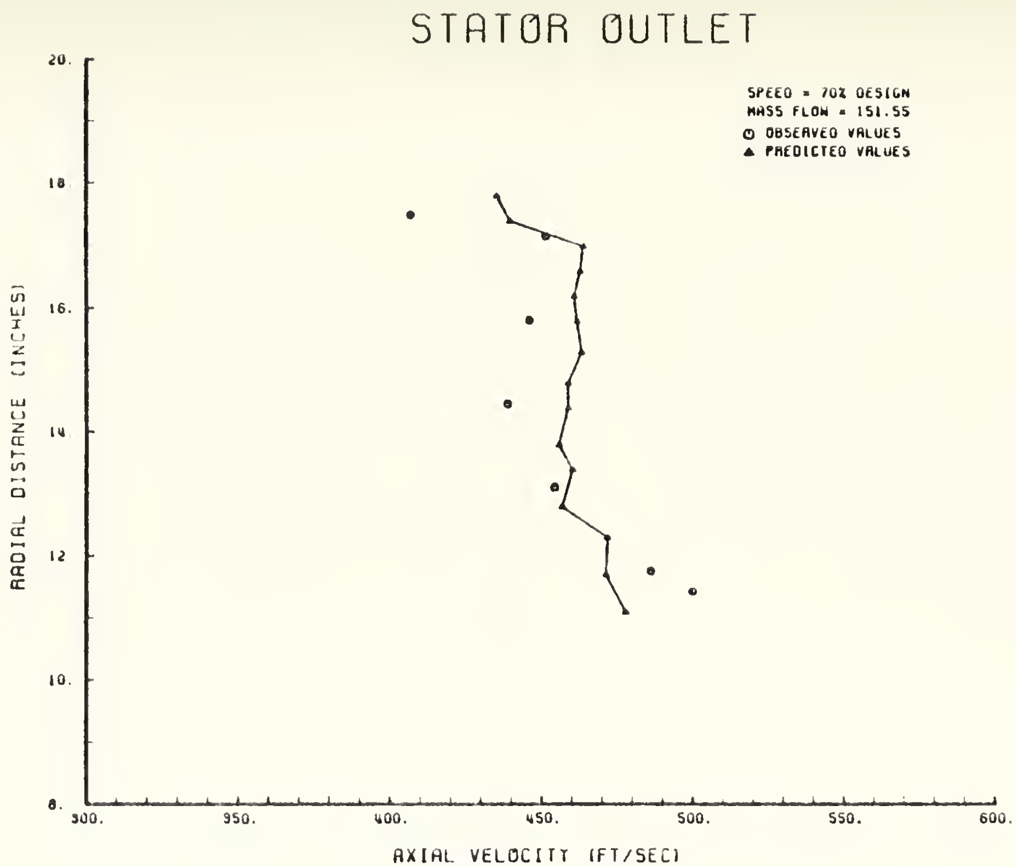

Figure 56. Axial Velocity at the Stator Outlet, 70% Design

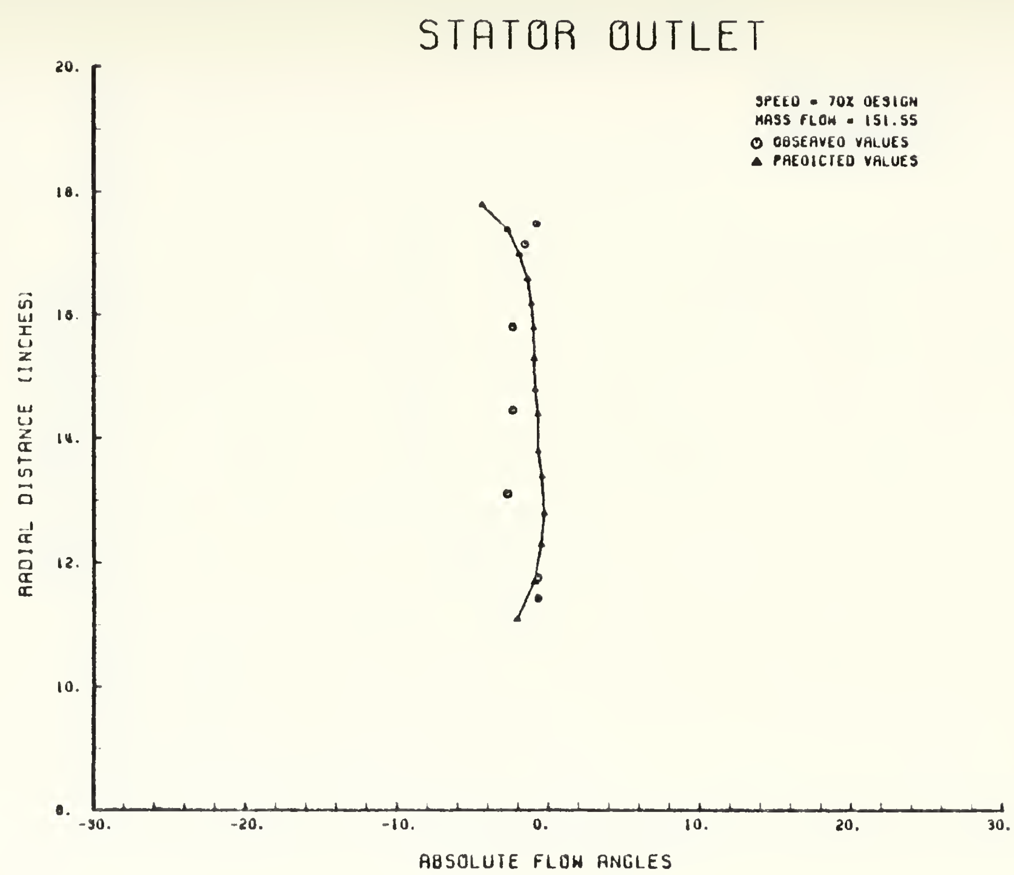

Figure 57. Absolute Flow Angles at Stator Outlet, 70% Design

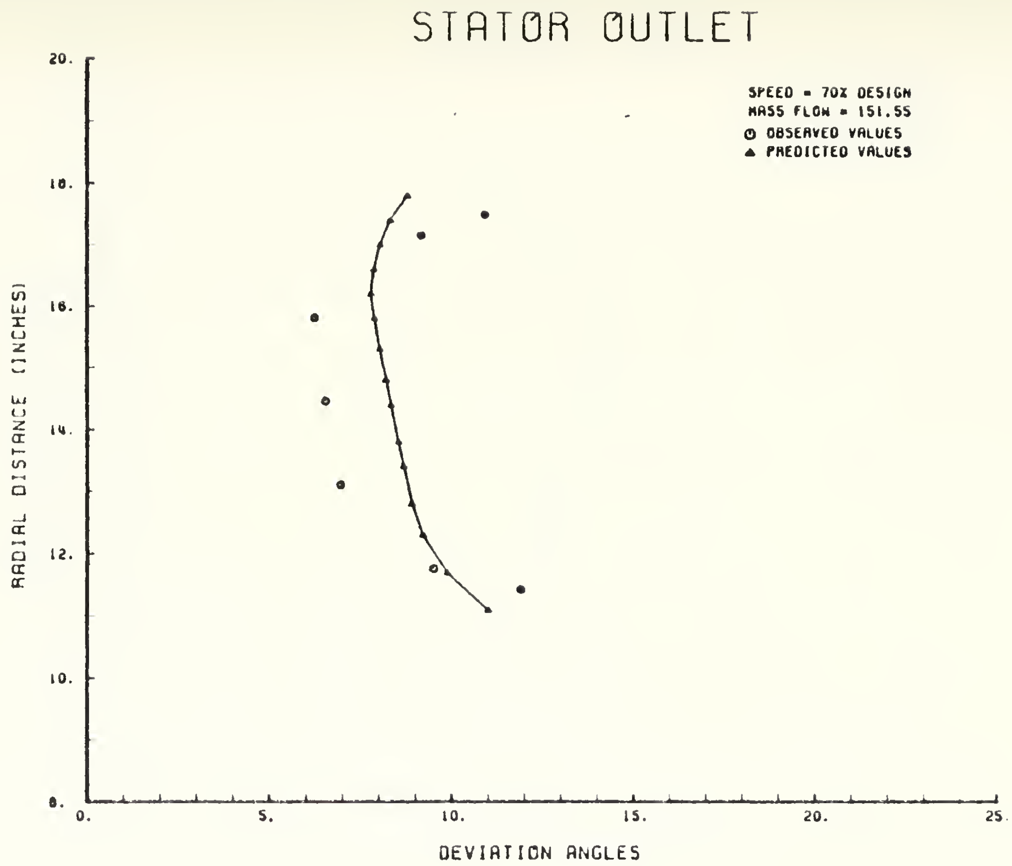

Figure 58. Deviation Flow Angles at Stator Outlet, 70% Design

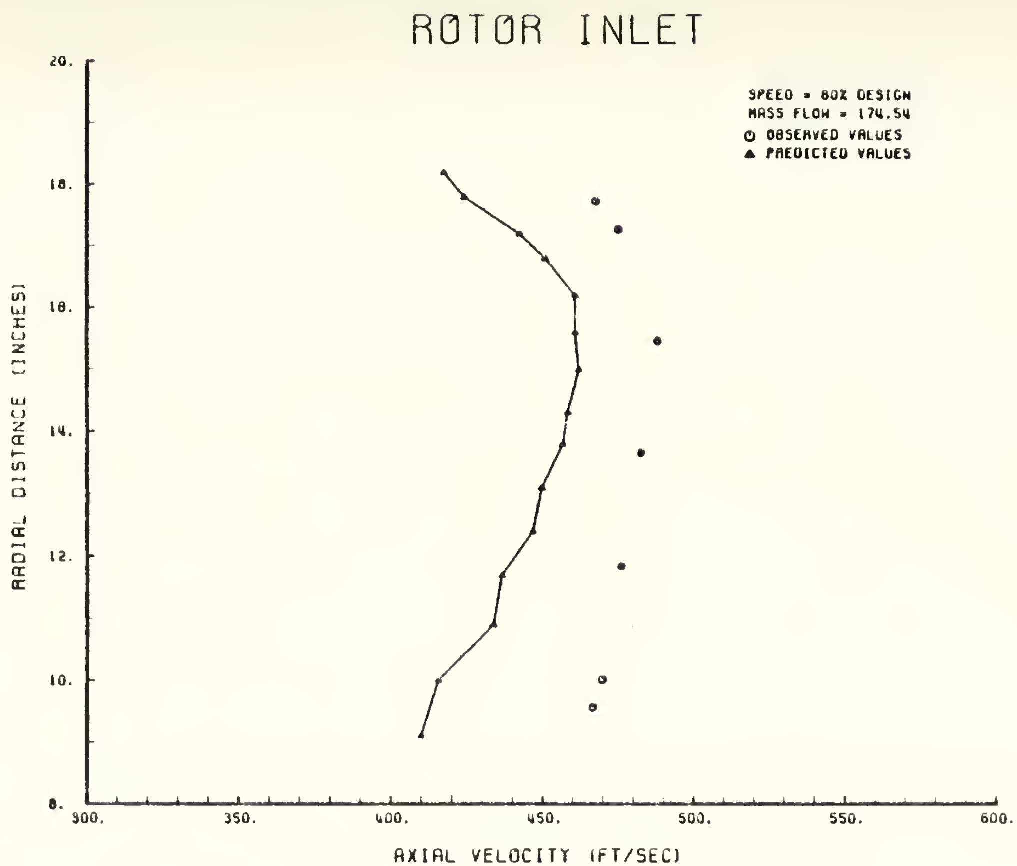

Figure 59. Axial Velocity at the Rotor Inlet, 80% Design

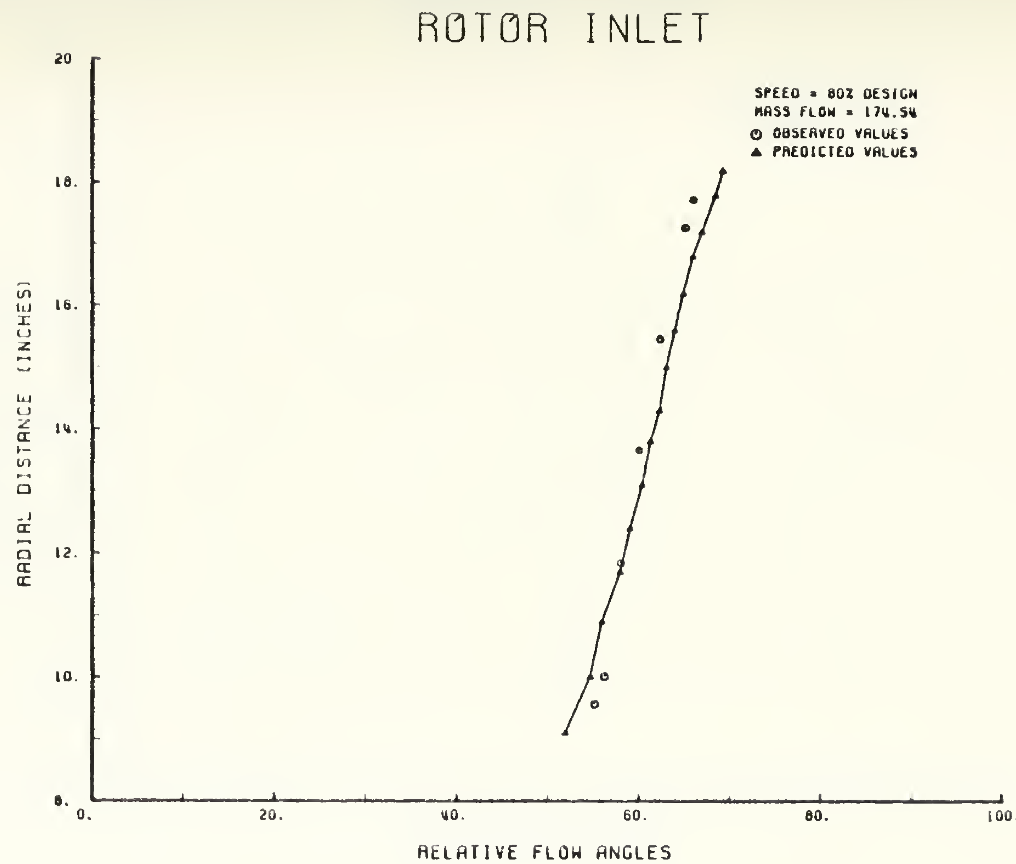

Figure 60. Relative Flow Angles at Rotor Inlet, 80% Design

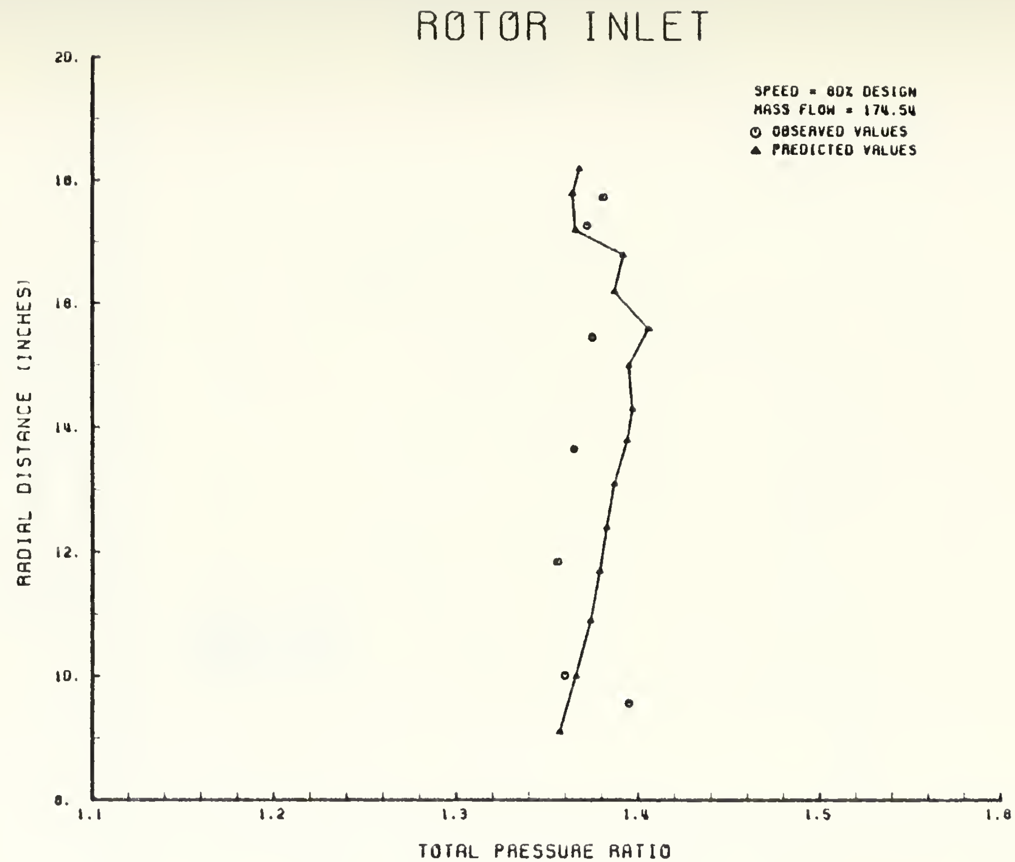

Figure 61. Total Pressure Ratio of the Rotor, 80% Design

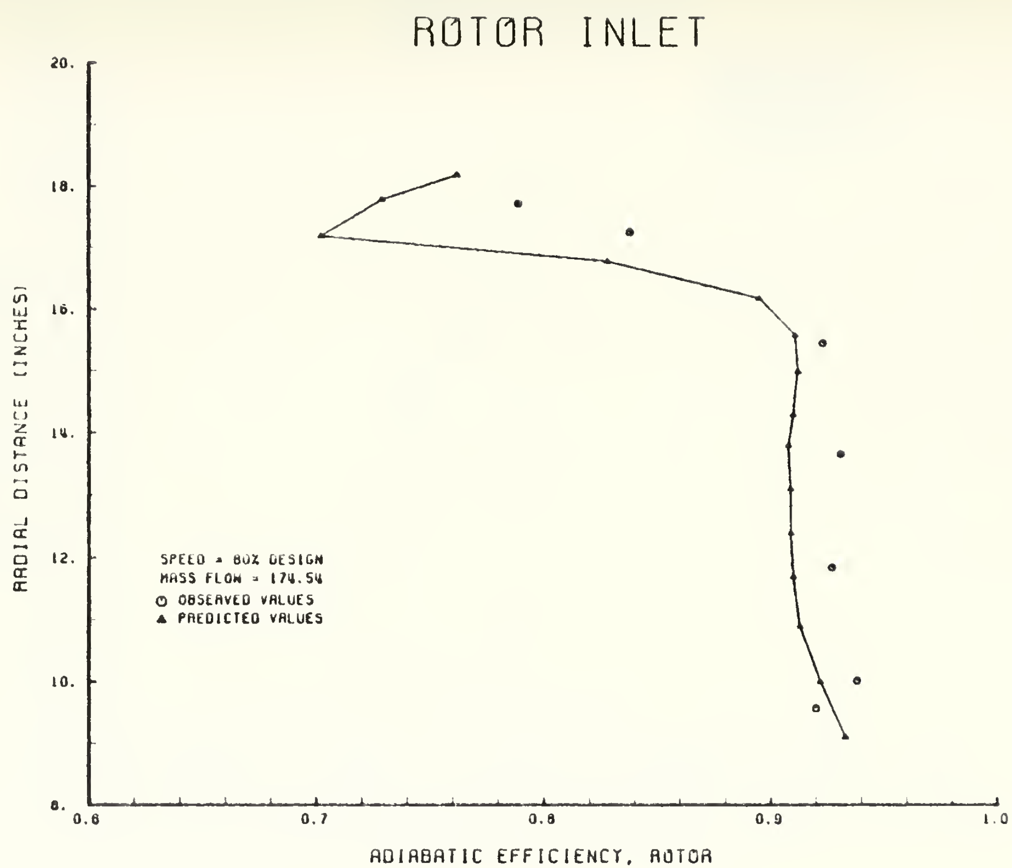

Figure 62. Adiabatic Efficiency of the Rotor, 80% Design

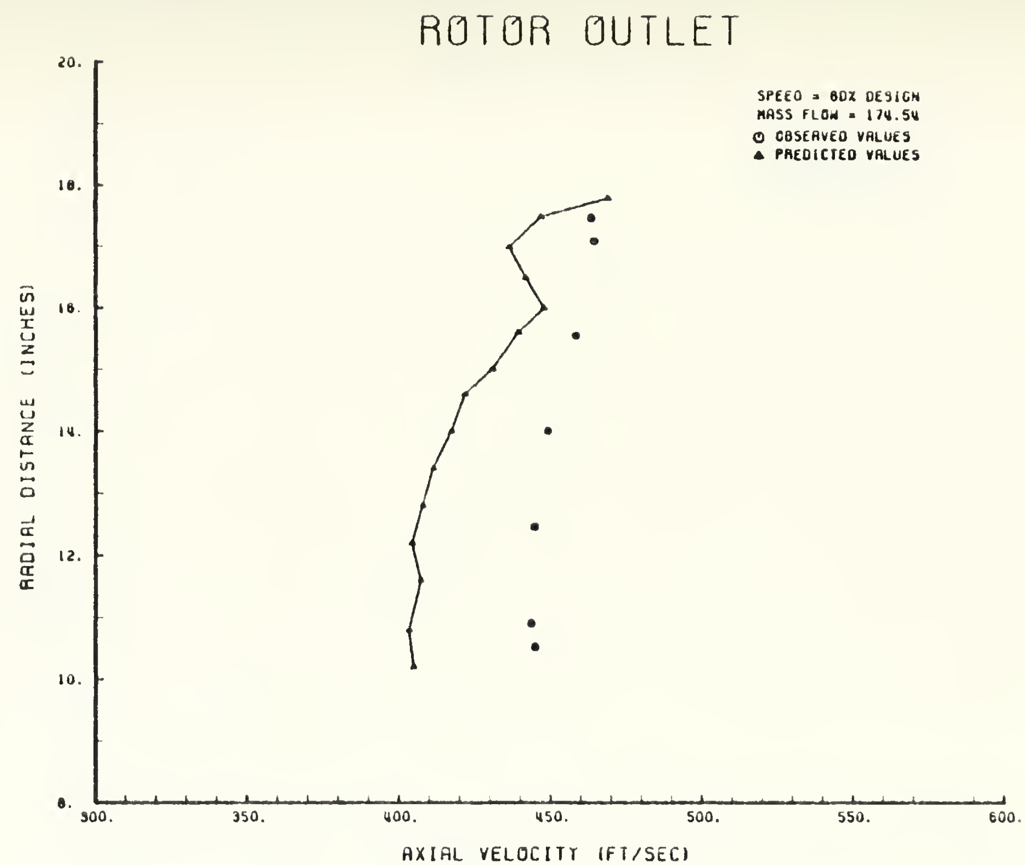

Figure 63. Axial Velocity at the Rotor Outlet, 80% Design

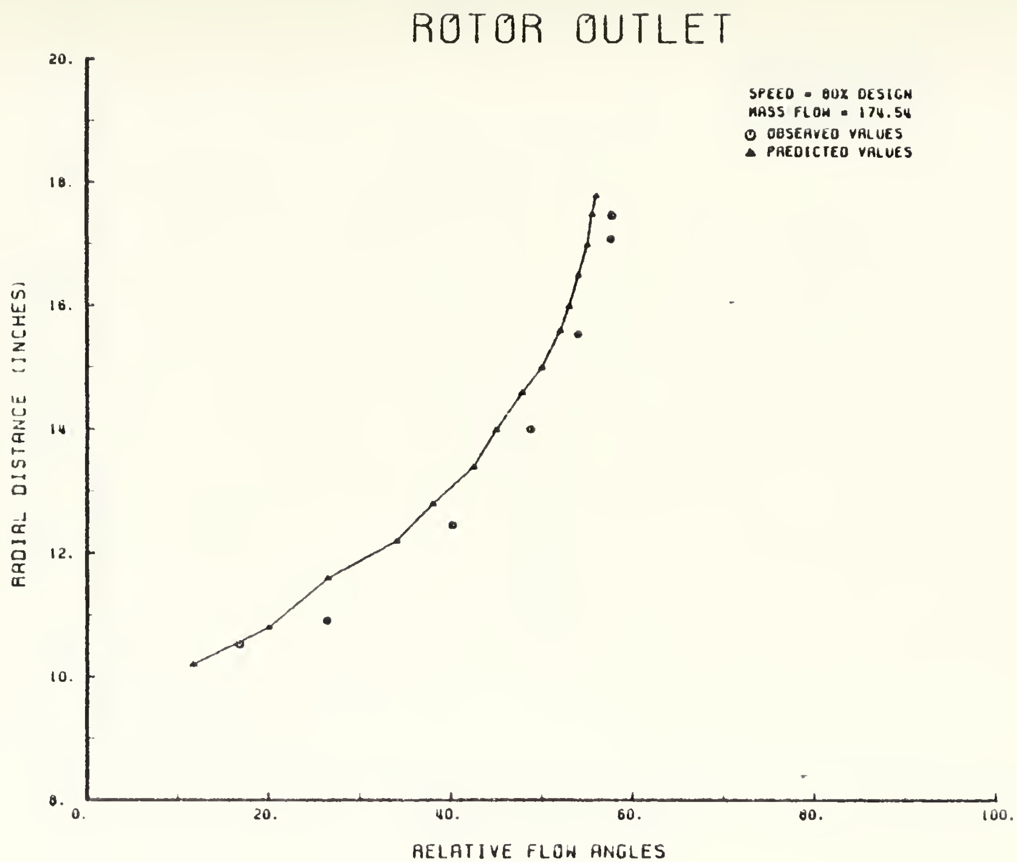

Figure 64. Relative Flow Angles at Rotor Outlet, 80% Design

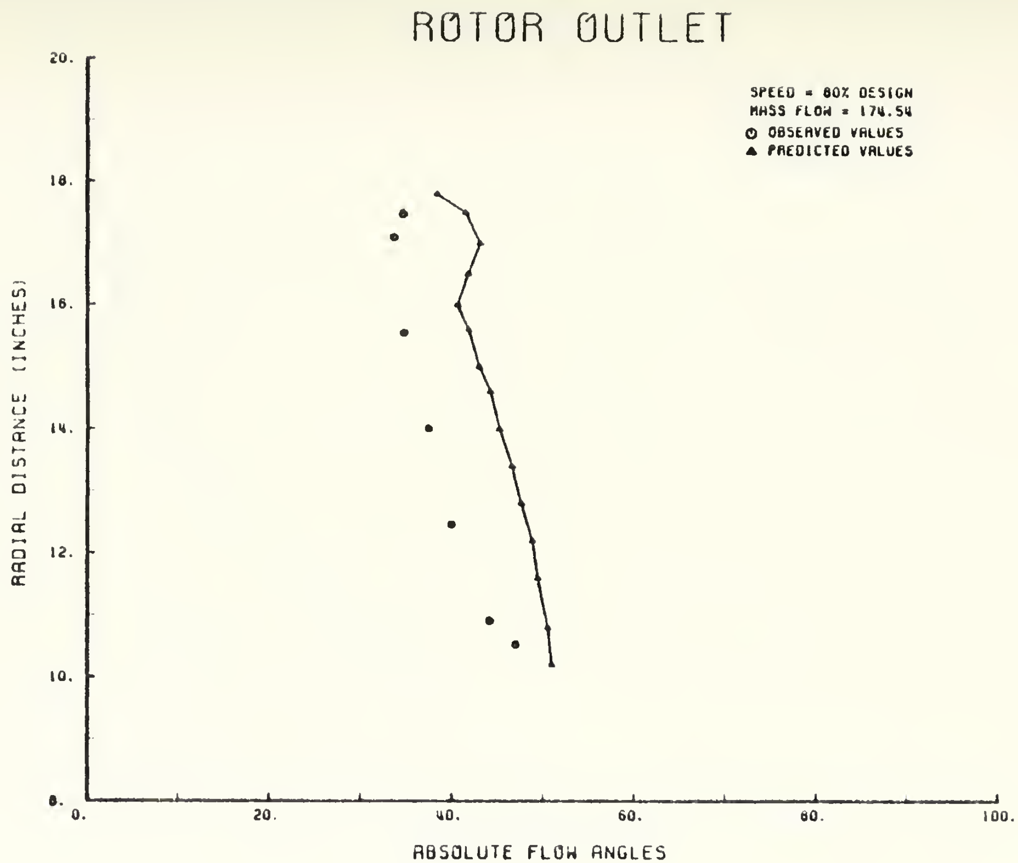

Figure 65. Absolute Flow Angles at Rotor Outlet, 80% Design

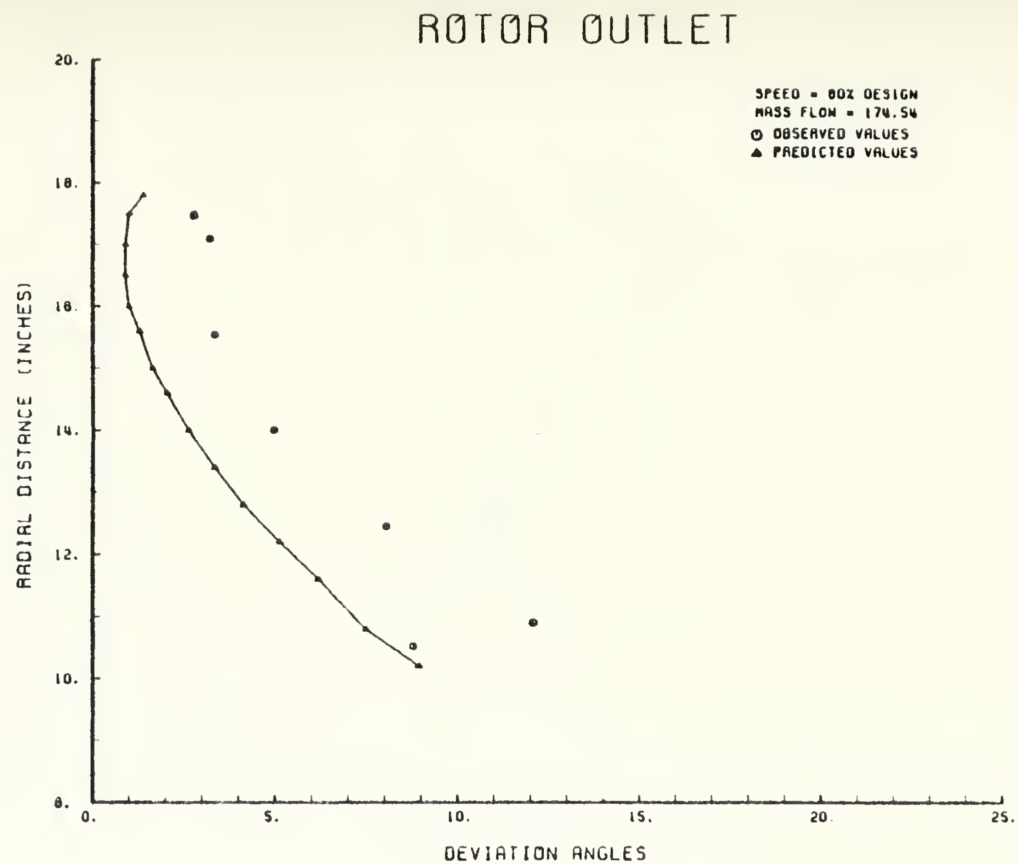

Figure 66. Deviation Flow Angles at Rotor Outlet, 80% Design

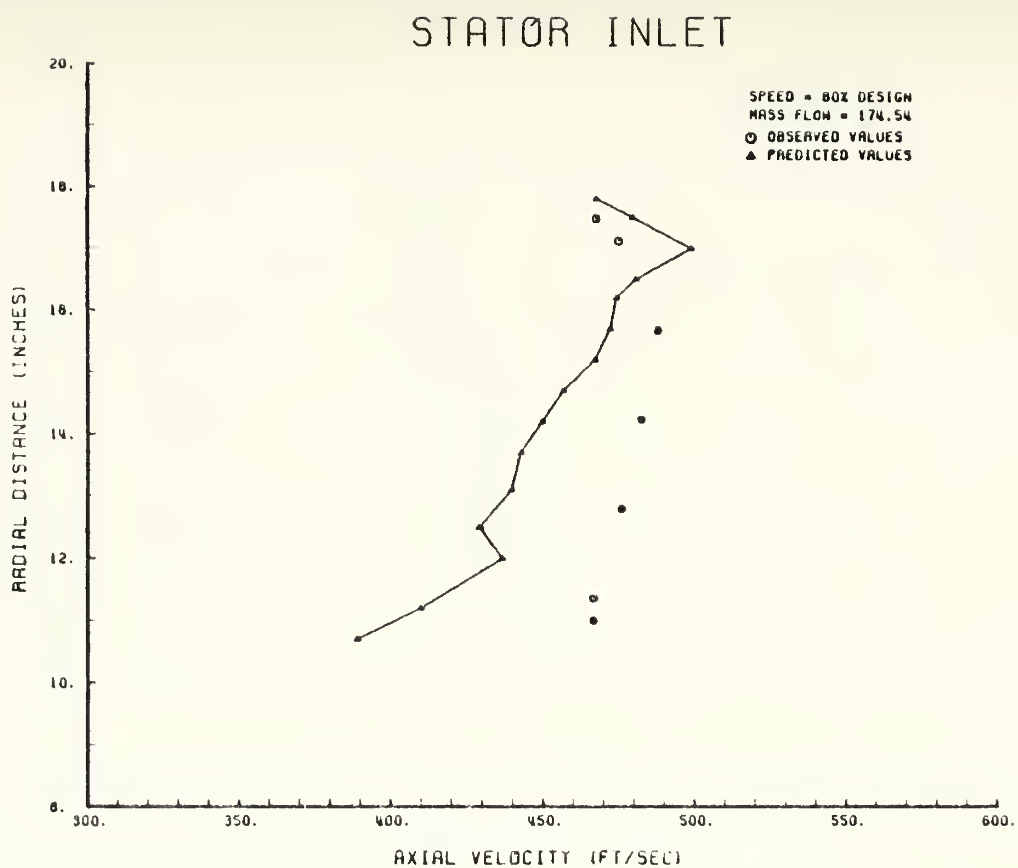

Figure 67. Axial Velocity at the Stator Inlet, 80% Design

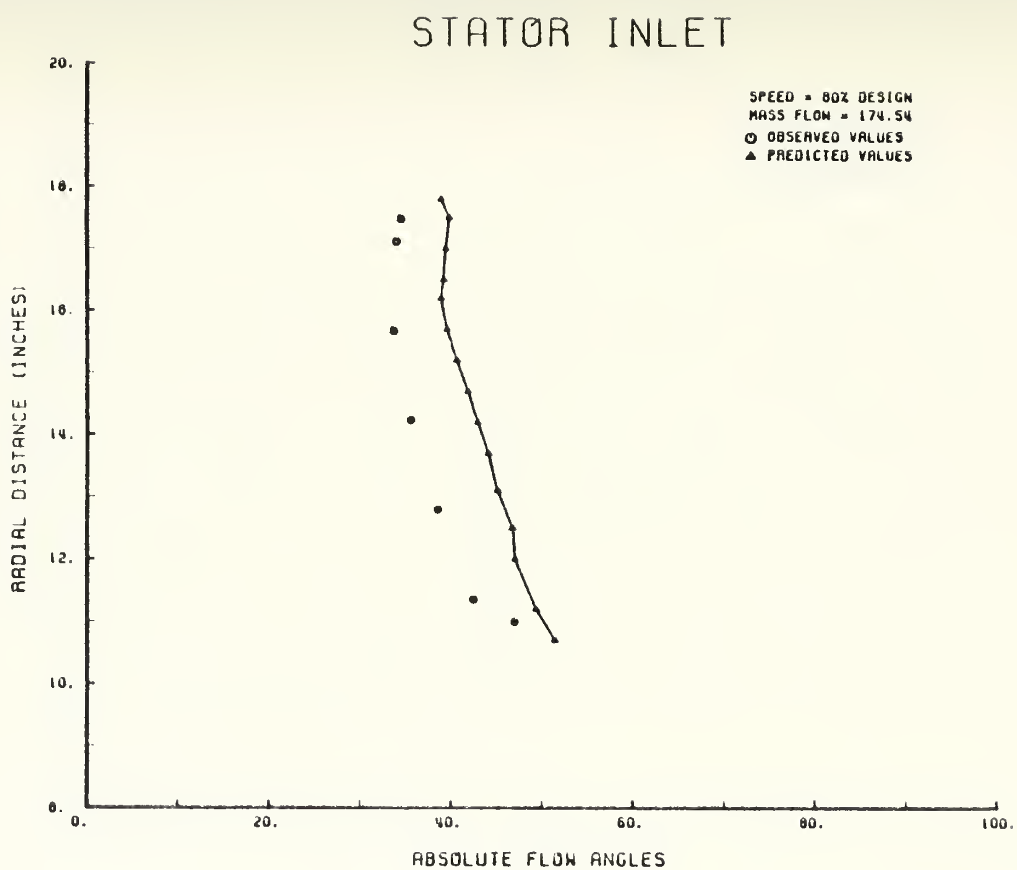

Figure 68. Absolute Flow Angles at Stator Inlet, 80% Design

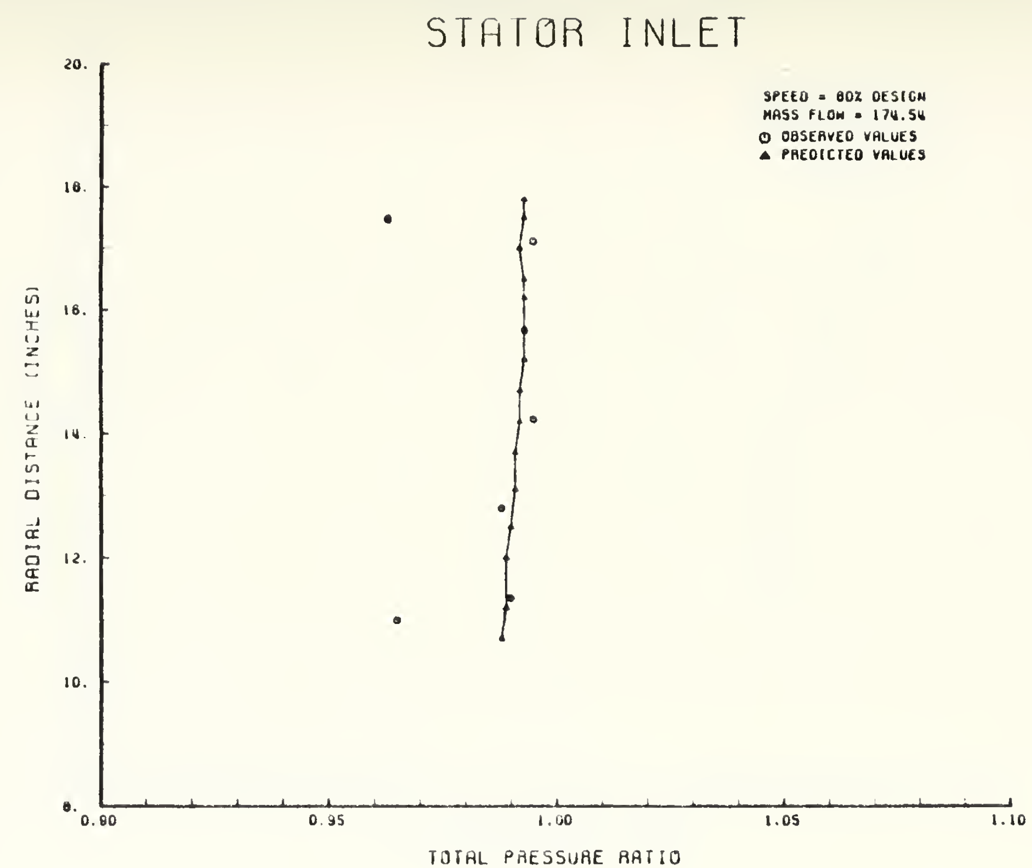

Figure 69. Total Pressure Ratio at Stator Inlet, 80% Design

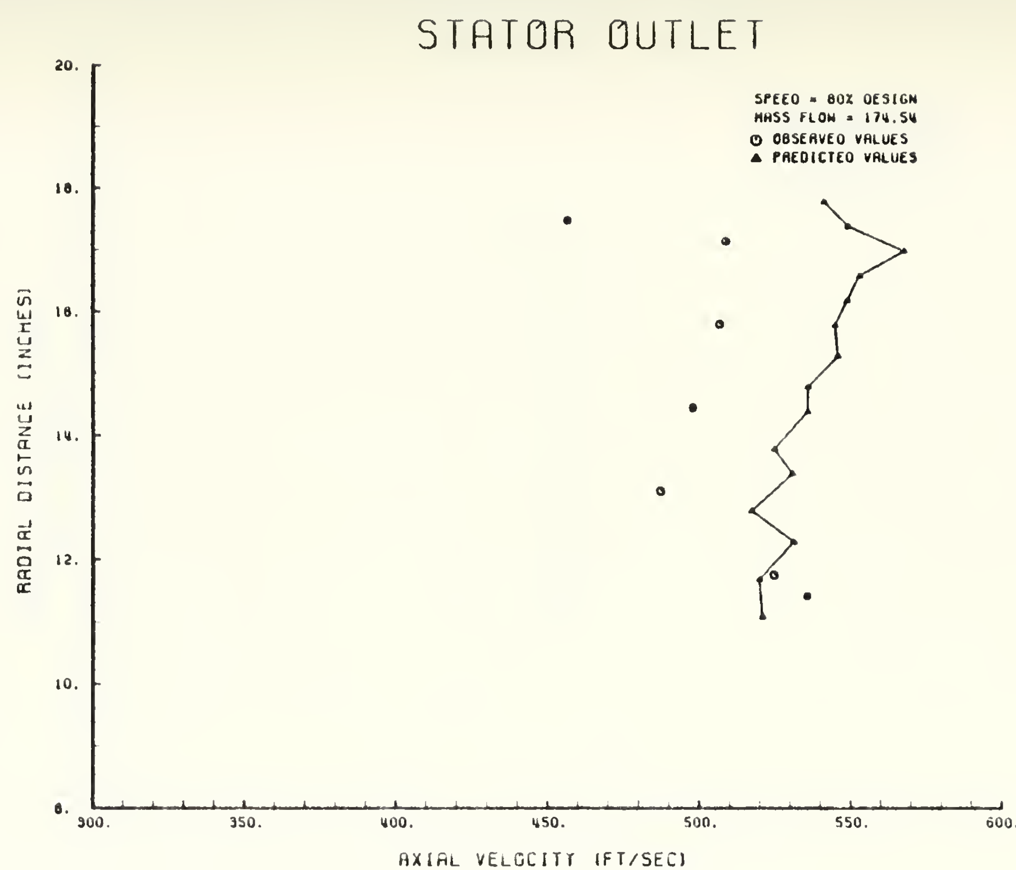

Figure 70. Axial Velocity at the Stator Outlet, 80% Design

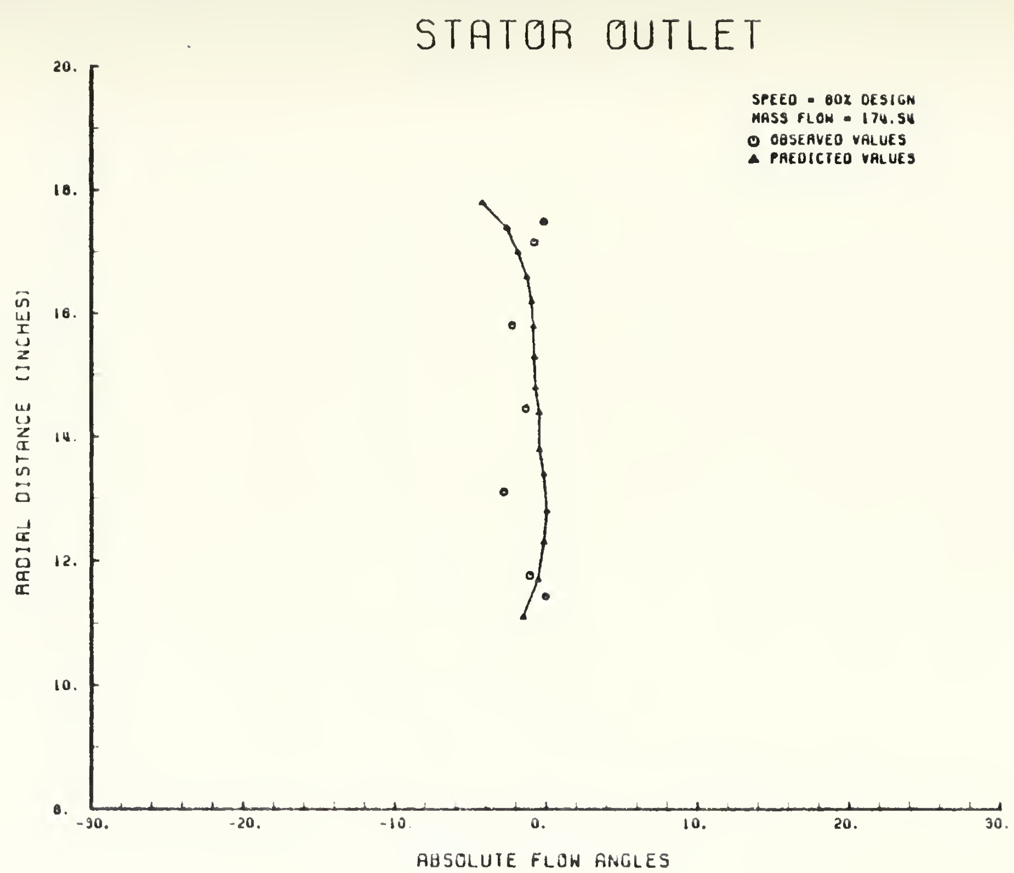

Figure 71. Absolute Flow Angles at Stator Outlet, 80% Design

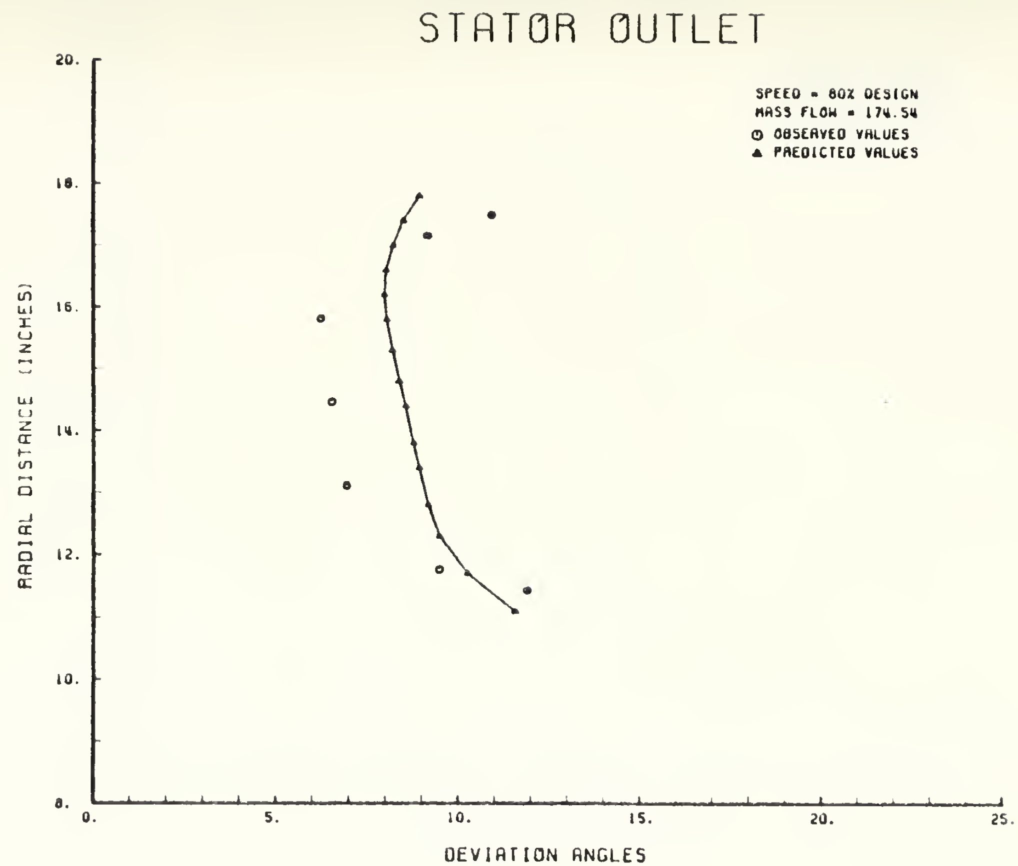

Figure 72. Deviation Flow Angles at Stator Outlet, 80% Design

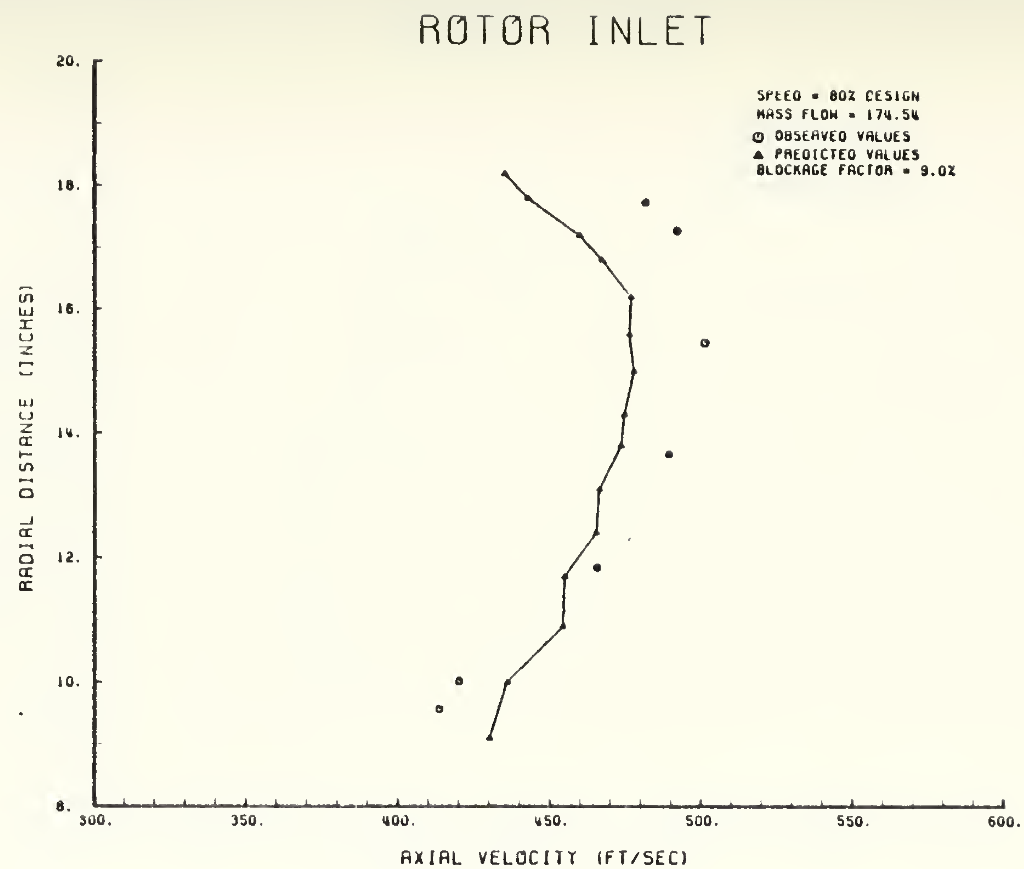

Figure 73. Axial Velocity at the Rotor Inlet, 80% Design, 9% Blockage

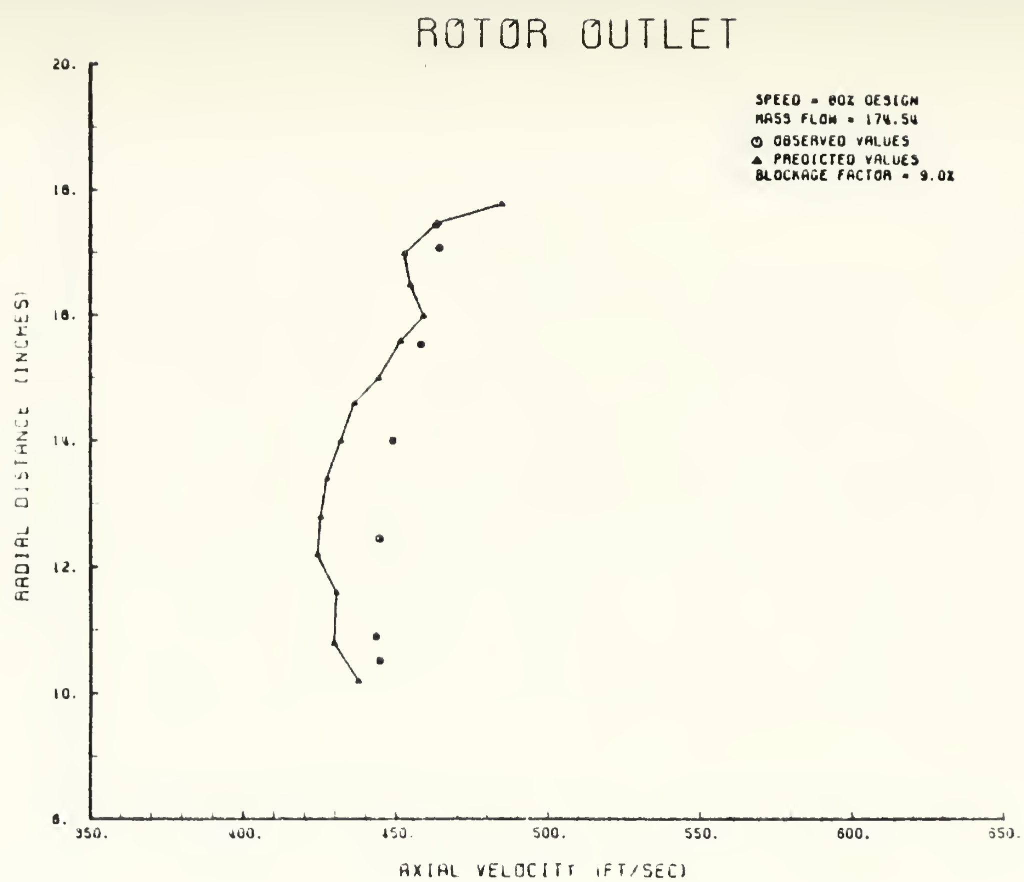

Figure 74. Axial Velocity at the Rotor Outlet, 80% Design, 9% Blockage

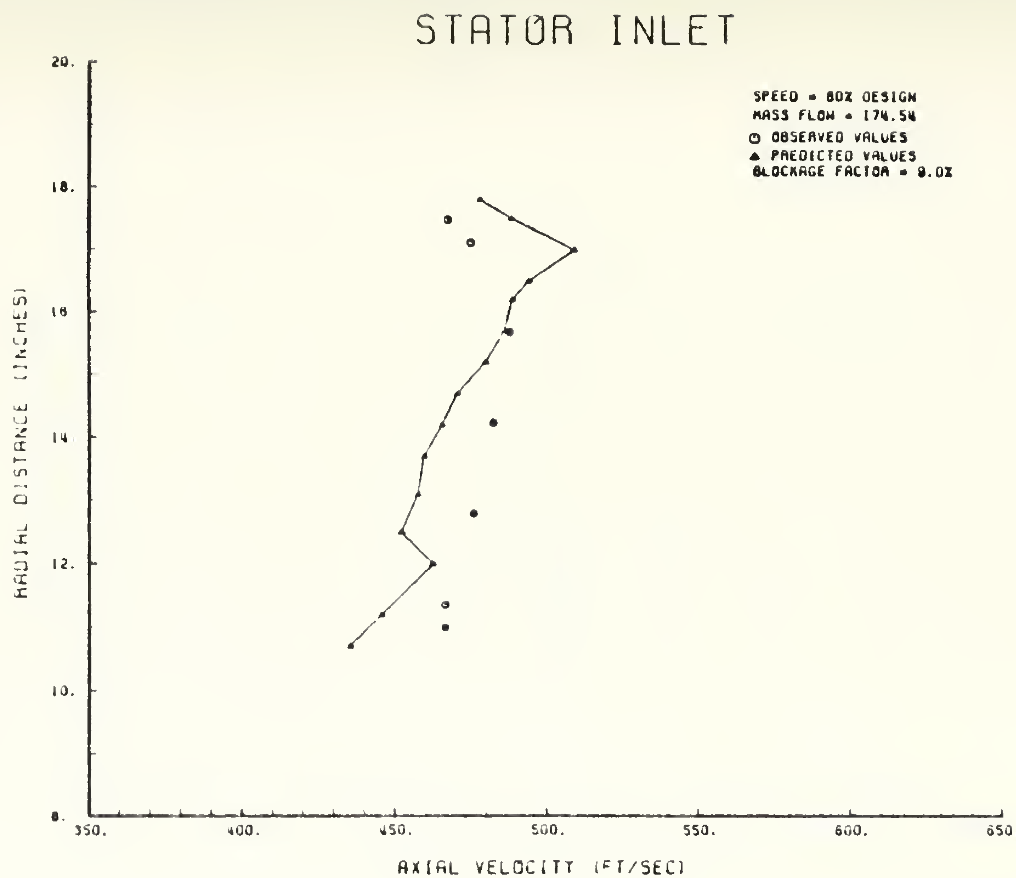

Figure 75. Axial Velocity at the Stator Inlet, 80% Design, 9% Blockage

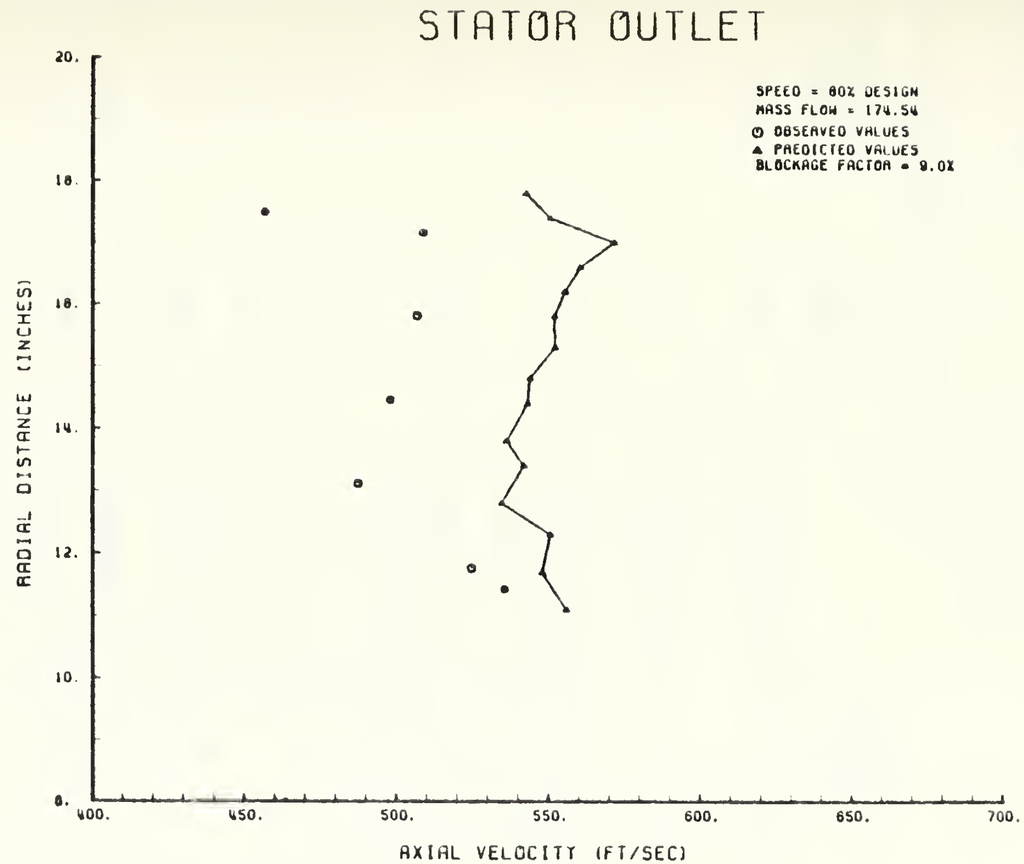

Figure 76. Axial Velocity at the Stator Outlet, 80% Design, 9% Blockage

APPENDIX A

STORAGE ALLOCATION FOR THE PROGRAM MESHGEN

The following list of pointers and variable names indicates the storage location of the first value of the corresponding array:

REAL 8 Variables (Array R8)

Pointer Name	Array Name	Array Contents
J1	XSE	Z Coordinates of the Super Elements
J2	YSE	R Coordinates of the Super Elements
J3	XXSE	Temporary Storage of the Z Coordinates of the Super Element Subdivision
J4	YYSE	Temporary Storage of the R Coordinates of the Super Element Subdivision
J5	ZC	Z Coordinates
J6	RC	R Coordinates
J7	PSI	Stream Function
J8	B	Blockage Factor
J9		End of Stored Values

REAL 4 Variables (Array O4)

Pointer Name	Array Name	Array Contents
M1	OHZC	Z Coordinates of Streamwise Element Boundaries
M2	OHRC	R Coordinates of Streamwise Element Boundaries
M3	OVZC	Z Coordinates of Transverse Element Boundaries
M4	OVRC	R Coordinates of Transverse Element Boundaries
M5	OHZC1	Z Coordinates of Streamwise Element Boundaries
M6	OHRC1	R Coordinates of Streamwise Element Boundaries
M7	OZC	Real 4 Z Coordinates of Mesh
M8	ORC	Real 4 R Coordinates of Mesh
M9		End of Stored Values

INTEGER 4 Variables (Array I2)

Pointer Name	Array Name	Array Contents
L1	NSEC	Number of Columns per Super Element
L2	NTSE	Super Element Type
L3	NTE	Element Type Identifier
L4	NODE	Connectivity Matrix
L5	NBC	Boundary Condition Nodes
L6		End of Stored Values

APPENDIX B

LISTING OF THE OUTPUT VARIABLES FOR THE PROGRAM MESHGEN

FORTRAN Name	Variable	File
ZC	Z Coordinate	TMESH DATA
RC	R Coordinate	TMESH DATA
B	Blockage Factor	TMESH DATA
PRESS	Inlet Static Pressure	FLUID DATA
PTOT	Inlet Total Pressure	FLUID DATA
TEMP	Inlet Static Temperature	FLUID DATA
TTOT	Inlet Total Temperature	FLUID DATA
RHOSTA	Inlet Static Density	FLUID DATA
RHOTT	Inlet Total Density	FLUID DATA
WDOT	Inlet Mass Flow	FLUID DATA
CP	Specific Heat	FLUID DATA
R	Gas Constant	FLUID DATA
G		FLUID DATA
RPM	RPM	FLUID DATA
VZI	Inlet Axial Velocity	FLUID DATA
NODE	Connectivity Matrix	NODE DATA
NTE	Element Type ID	NODE DATA

NBC	Boundary Nodes Where ψ Specified	BC DATA
PSI	Stream Function	STREAM DATA

APPENDIX C

STORAGE ALLOCATION FOR THE PROGRAM TURBO

The following list of pointers and variable names indicates the storage location of the first value of the corresponding array:

REAL 8 Variables (Array R8)

Pointer Name	Array Name	Array Contents
NP1	ZC	Z Coordinates
NP2	RC	R Coordinates
NP3	B	Blockage Factors
NP4	ALP	Absolute Flow Angles
NP5	BE	Relative Flow Angles
NP6	H	Total Enthalpy
NP7	HS	Static Enthalpy
NP8	VZ	Axial Velocity
NP9	VR	Radial Velocity
NP10	VU	Absolute Tangential Velocity
NP11	WU	Relative Tangential Velocity
NP12	PSI	Current Streamfunction Values
NP13	PSIO	Previous Streamfunction Value
NP14	F	Right-hand Side Vector
NP15	RHS	Temporary Storage for F
NP16	RHO	Static Density

NP17	RHON	Not Used
NP18	WRL	Angular Momentum Vector
NP19	ETA	Adiabatic Efficiencies
NP20	EM	Stiffness Matrix Elements
NP21	DEV1	Rotor/Stator Deviation Angles
NP22	PRAT	Total-to-Total Pressure Ratio
NP23	TEMP	Static Temperature
NP24	TTOT	Total Temperature
NP25	PRESS	Static Pressure
NP26	PTOT	Total Pressure
NP27	RHOTT	Total Density
NP28	HR	Rhothalpy
NP29	ENTROP	Entropy
NP30		End of Stored Values

REAL 4 Variables (Array O4)

Pointer Name	Array Name	Array Contents
NPO1	OVEL	Velocity at a Given Station
NPO2	ORC	R Coordinates
NPO3	OBE	Relative Flow Angles
NPO4	OALP	Absolute Flow Angles
NPO5		End of Stored Values

INTEGER 4 Variables (Array I2)

Pointer Name	Array Name	Array Contents
NPI1	NFS	Nodes for F Calculation
NPI2	NBC	Boundary Nodes
NPI3	NTE	Element Type Identifier
NPI4	NODE	Connectivity Matrix
NPI5		End of Stored Values

APPENDIX D

LISTING OF THE OUTPUT VARIABLES FOR THE PROGRAM TURBO

Listing Name	Variable	Units
PSI	Stream Function	lbm/sec
VZ	Axial Velocity	ft/sec
VR	Radial Velocity	ft/sec
R	Radius	inches
DENSITY	Static Density	lbm/ft^3
WRL	Angular Momentum	ft^2/sec
HT	Total Enthalpy	BTU/lbm
VT	Absolute Tangential Velocity	ft/sec
WT	Relative Tangential Velocity	ft/sec
HS	Static Enthalpy	BTU/lbm
TEMP	Static Temperature	°R
TTOT	Total Temperature	°R
PRESS	Static Pressure	psia
PTOT	Total Pressure	psia
RHOT	Total Density	lbm/ft^3
ALPHA	Absolute Flow Angle	Degrees
BETA	Relative Flow Angle	Degrees

APPENDIX E

LISTING OF THE PROGRAM MESHGEN

FILE: MESHGEN FORTRAN A1 NAVAL POSTGRADUATE SCHOOL

```
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                 ++
C ++                      PROGRAM MESHGEN                            ++
C ++                                                                 ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                 ++
C ++   THIS PROGRAM PRODUCES A RECTANGULAR EIGHT-NODED                ++
C ++   ISOPARAMETRIC GRID OF ARBITRARY DIMENSION M X N GIVEN         ++
C ++   THE CORNER NODES OF THE MESH'S "SUPER ELEMENTS".              ++
C ++   THE PROGRAM ALSO COMPUTES AN INITIAL STREAM FUNCTION          ++
C ++   DISTRIBUTION, TANGENTIAL BLOCKAGE FACTORS AND THE             ++
C ++   APPROPRIATE INLET CONDITIONS NEED FOR THE PROGRAM TURBO.      ++
C ++   THE PROGRAM IS ONLY LIMITED BY THE SPACE ALLOCATED IN         ++
C ++   THE ARRAYS R8, C4, I2 OR MACHINE LIMITATIONS.                 ++
C ++                                                                 ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-G,P-Z), REAL*4(H)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      INTEGER*4 IANS
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /NCOUNT/ MROW,MROW1,MNSEC
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      COMMON /NPOINT/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,
     1L1,L2,L3,L4,L5,L6,M1,M2,M3,M4,M5,M6,M7,M8,M9
      COMMON /RPAR/ R8(5000)
      COMMON /NPAR/ I2(3000)
      COMMON /CPAR4/ C4(3000)
      LIMR = 5000
      LIMI = 3000
      LIM4 = 3000
C
C
      CALL INIT1(MROW2,NCOL2,NSE,NSE2,WDOT,PTOT,TTOT,RPM,G,R,CP,ZMAX,
     1ZMIN,RMAX,RMIN,IANS)
C
C
      CALL INPUT(R8(J1),R8(J2),I2(L1),I2(L2),NE1,NSE2,NSE,MRR1,N2,LIMR,
     1LIMI,LIM4,IANS)
C
C
      CALL TMESH(R8(J1),R8(J2),R8(J3),R8(J4),R8(J5),R8(J6),I2(L1)
     1,I2(L2),I2(L3),R8(J13),NSE2,NSE,N2,MRR1,IANS)
C
C
      CALL CONEC(I2(L2),I2(L3),I2(L4),NSE,N2,NE1,NROTOB,NROTOE,
     1NSTATB,NSTATE,IANS)
C
C
      CALL INIT2(R8(J6),I2(L4),I2(L5),R8(J7),RHOSTA,RHOTT,TTOT,WDOT,
     1PTOT,PRESS,TEMP,VZI,R,CP,G,NSE2,MRR1,N2,NSE,NE1,IANS)
C
C
      IF(IANS.EQ.2) GOTO 100
C
C
      CALL TASK1(R8(J6),R8(J13),NSE2,MRR1,N2,NROTOB,NROTOE,NSTATB,
     1NSTATE)
C
C
100   CALL FILGEN(R8(J5),R8(J6),R8(J13),R8(J7),I2(L4),I2(L3),I2(L5),
     1PRESS,PTOT,TEMP,TTOT,RHOSTA,RHOTT,VZI,WDOT,RPM,CP,R,G,NROTOB,
     2NSTATB,NSE2,NSE,N2,MRR1,NE1,IANS)
C
      WRITE(15,200)
200   FORMAT(5X,' >DO YOU WANT A PLOT OF THE MESH ?',/,'  1 = YES
     1  2 = NO')
```

```
        READ(15,*) IANS
      IF(IANS.EQ.2) GOTO 300
C
      CALL MPLOT(R8(J5),R8(J6),ZMAX,ZMIN,RMAX,RMIN,C4(M1),C4(M2),
     1C4(M3),C4(M4),C4(M5),C4(M6),C4(M7),C4(M8),I2(L5),NSE2,MRR1,
     2N2,NSE,NE1,NC2,MM2)
C
C
300   STOP
      END
C
C
C
      SUBROUTINE INIT1(MROW2,NCOL2,NSE,NSE2,WDOT,PTOT,TTOT,RPM,G,R,CP,
     1ZMAX,ZMIN,RMAX,RMIN,IANS)
      IMPLICIT REAL*8(A-G,P-Z), REAL*4(H)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      INTEGER*4 IANS
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /MCOUNT/ MROW,MROW1,MNSEC
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      COMMON /NPOINT/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,
     1L1,L2,L3,L4,L5,L6,M1,M2,M3,M4,M5,M6,M7,M8,M9
      WRITE(15,100)
100   FORMAT(5X,' >ENTER INLET CONDITIONS:',/,'   MASS FLOW(LBM PER SEC)
     1 TOTAL TEMP(DEG R) AND TOTAL PRESS(PSIA)'/)
        READ(15,*) WDOT,TTOT,PTOT
      WRITE(15,110)
110   FORMAT(5X,' >ENTER OPERATING CONSTANTS:',/,'   RPM',/,'   RATIO OF
     1 SPECIFIC HEATS (GAMMA)',/,'   GAS CONSTANT R(FT-LBF/LBM-DEG R)',/
     2,'   SPECIFIC HEAT CONSTANT PRESSURE CP(BTU/LBM-DEG R)',/)
      READ(15,*) RPM,G,R,CP
      WRITE(15,120)
120   FORMAT(5X,' >ENTER GRAPH SCALING CONSTANTS:',/,'   Z MAX',/,'   Z
     1MIN',/,'   R MAX',/,'   R MIN',/)
      READ(15,*) ZMAX,ZMIN,RMAX,RMIN
        WRITE(15,50)
50      FORMAT(5X,' >DO YOU WANT TO CREATE A NEW MESH ?',/,'  1 = YES
     1 2 = NO')
        READ(15,*) IANS
      IF(IANS.EQ.1) GOTO 199
      READ(25,130) NN,MROW,NCOL,MROW1,NCOL1,NROTCB,NSTATB
130   FORMAT(7I5)
      GOTO 201
199   WRITE(15,200)
200   FORMAT(5X,' >ENTER NO OF SUPER ELEMENTS AND NO ELEMENT ROWS',/)
      READ(15,*) NSE,MROW
201   MROW1 = 2 * MROW + 1
      MR1 = MROW1 - 1
      MR = MROW
      MROW2 = MROW1
      NCOL2 = NCOL1
      NSE2 = 2*NSE + 2
      J1 = 1
      J2 = J1 + NSE2
      J3 = J2 + NSE2
      L1 = 1
      L2 = L1 + NSE
      L3 = L2 + NSE
      RETURN
      END
C
C
      SUBROUTINE INPUT(XSE,YSE,NSEC,NTSE,NE1,NSE2,NSE,MRR1,N2,LIM3,
     1LIM1,LIM4,IANS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++  THIS SUBROUTINE RETAINS THE DISCRIPTION OF THE SUPER ELEMENTS   ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      INTEGER*4 IANS
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /MCOUNT/ MROW,MROW1,MNSEC
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      COMMON /NPOINT/ J1,J2,J3,J4,J5,J6,J7,J8,J9,J10,J11,J12,J13,J14,
     1L1,L2,L3,L4,L5,L6,M1,M2,M3,M4,M5,M6,M7,M8,M9
      DIMENSION XSE(1),YSE(1)
      DIMENSION NSEC(1),NTSE(1)
C
C
      IF(IANS.EQ.2) GOTO 202
      NCOL = 0
      J = 1
      NSE22 = NSE2/2
      DO 100 I = 1,NSE22
        WRITE(15,101) I
101     FORMAT(5X,' >ENTER Z,R COORDINATE PAIRS FOR STATION ',I2/)
        READ(15,*) XSE(J),YSE(J),XSE(J+1),YSE(J+1)
        J = J + 2
100   CONTINUE
      DO 200 I = 1,NSE
        WRITE(15,201) I
201     FORMAT(5X,' >ENTER TYPE OF SUPER ELEMENT AND THE NO OF COLUMNS F
     1OR SUPER ELEMENT ',I2/)
        READ(15,*) NTSE(I),NSEC(I)
        NCOL = NCOL + NSEC(I)
200   CONTINUE
202   NE = MROW * NCOL
      NE1 = NE
      NCOL1 = 2 * NCOL + 1
      NN = NCOL1 * MROW1 - MROW * NCOL
      N2 = NN
      MRR1 = MROW1
      J1 = 1
      J2 = J1 + NSE2
      J3 = J2 + NSE2
      J4 = J3 + MRR1
      J5 = J4 + MRR1
      J6 = J5 + N2
      J7 = J6 + N2
      J8 = J7 + N2
      J9 = J8 + N2
      J10 = J9 + N2
      J11 = J10 + N2
      J12 = J11 + N2
      J13 = J12 + N2
      J14 = J13 + N2
      L1 = 1
      L2 = L1 + NSE
      L3 = L2 + NSE
      L4 = L3 + NE1
      L5 = L4 + 8*NE1
      L6 = L5 + NE1
      M1 = 1
      M2 = M1 + NCOL1
      M3 = M2 + NCOL1
      M4 = M3 + MRR1
      M5 = M4 + MRR1
      M6 = M5 + NCOL1
      M7 = M6 + NCOL1
      M8 = M7 + N2
      M9 = M8 + N2
      NEXCR = LIMR - J14
      NEXCI = LIMI - L6
      NEXC4 = LIM4 - M9
      IF(NEXCR.LT.0) CALL ERR1(NEXCR)
      IF(NEXCI.LT.0) CALL ERR2(NEXCI)
      IF(NEXC4.LT.0) CALL ERR3(NEXC4)
      WRITE(6,300) NEXCR,NEXCI
```

```
300   FORMAT(10X,' MEMORY SPACE AVAILABLE : REAL =',I5,2X,' INTEGER = '
     1,I5///)
      RETURN
      END
C
C
C
      SUBROUTINE TMESH(XSE,YSE,XXSE,YYSE,ZC,RC,NSEC,NTSE,NTE,
     1B,NSE2,NSE,N2,MRR1,IANS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                 ++
C ++   THIS SUBROUTINE PRODUCES A RECTANGULAR EIGHT-NODED             ++
C ++   ISOPARAMETRIC GRID OF ARBITRARY DIMENSION M X N GIVEN         ++
C ++   THE CORNER NODES OF THE MESH'S "SUPER ELEMENTS"               ++
C ++                                                                 ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      INTEGER*4 IANS
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /MCOUNT/ MROW,MROW1
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      DIMENSION XSE(1),YSE(1),XXSE(1),YYSE(1),ZC(1),RC(1),B(1)
      DIMENSION NSEC(1),NTSE(1),NTE(1)
C
      IF(IANS.EQ.1) GOTO 20
      DO 11 I=1,NN
        READ(20,12) ZC(I),RC(I),B(I)
12      FORMAT(3F15.11)
11    CONTINUE
      GOTO 100
20    YDIV1 = FLOAT(MR1)
      YDIV2 = FLOAT(MR)
      I = 1
      IJ = 1
      IL = 1
      IM = 1
      NSTAT = 0
      NROTC = 0
      DO 10 N = 1,NN
        ZC(N) = 0.D0
        RC(N) = 0.D0
10    CONTINUE
      DO 100 J = 1,NSE
        IF(J.EQ.1) GOTO 120
        IF(NTSE(J)-NTSE(J-1)) 120,120,140
140     IF(IM.LT.0) GOTO 150
        NROTC = IL
        IM = 0 - IM
        GOTO 120
150     NSTAT = IL
        IM = 0 - IM
120     IT = NSEC(J) * MROW
        DO 110 L = 1,IT
          NTE(IL) = NTSE(J)
          IL = IL + 1
110     CONTINUE
        NNCOL1 = 2 * NSEC(J) + 1
        NC1 = NNCOL1
        NC = NC1 - 1
        XDIV1 = FLOAT(NC1)
        XDIV2 = FLOAT(NC)
        X1 = XSE(I)
        X2 = XSE(I + 1)
        X3 = XSE(I + 2)
        X4 = XSE(I + 3)
        Y1 = YSE(I)
        Y2 = YSE(I + 1)
        Y3 = YSE(I + 2)
```

```
          Y4 = YSE(I + 3)
          XDIF2 = X3 - X1
          XDIF3 = X4 - X2
          YDIF2 = Y1 - Y3
          YDIF3 = Y2 - Y4
          XINT2 = XDIF2 / XDIV2
          XINT3 = XDIF3 / XDIV2
          YINT2 = YDIF2 / XDIV2
          YINT3 = YDIF3 / XDIV2
          ISEL = -1
          DO 200 K = 1,NC1
             XDIF1 = X1 - X2
             YDIF1 = Y1*Y1 - Y2*Y2
             IF(ISEL) 210,210,220
210          XINT1 = XDIF1 / YDIV1
             YINT1 = YDIF1 / YDIV1
             MRR = MROW1
             GOTO 230
220          XINT1 = XDIF1 / YDIV2
             YINT1 = YDIF1 / YDIV2
             MRR = MROW + 1
230          DO 300 IC = 1, MRR
                IC1 = IC - 1
                XXSE(IC) = X1 - FLOAT(IC - 1)*XINT1
                IF(IC.NE.1) GOTO 231
                YYSE(IC) = Y1
                GOTO 232
231             YYSE(IC) = DSQRT(YYSE(IC1)*YYSE(IC1) - YINT1)
232             ZC(IJ) = XXSE(IC)
                RC(IJ) = YYSE(IC)
                IJ = IJ + 1
300          CONTINUE
             X1 = X1 + XINT2
             X2 = X2 + XINT3
             Y1 = Y1 - YINT2
             Y2 = Y2 - YINT3
             ISEL = -ISEL
200       CONTINUE
          ISEL = -ISEL
          IJ = IJ - MROW1
          I = 2*J + 1
100    CONTINUE
       WRITE(6,400) MROW,NCOL
400    FORMAT(10X,' NUMBER OF ROWS = ',I3,18X,' NUMBER OF COLUMNS = '
      1,I3/)
       WRITE(6,410) NE,NN
410    FORMAT(5X,' TOTAL NUMBER OF ELEMENTS = ',I4,10X,
      1' TOTAL NUMBER OF NODES = ',I4//)
       RETURN
       END
C
C
       SUBROUTINE CONEC(NTSE,NTE,NODE,NSE,N2,NE1,NROTOB,NROTOE,NSTATB,
      1NSTATE,IANS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                               ++
C ++         THIS SUBROUTINE PRODUCES THE CONECTIVITY MATRIX        ++
C ++                                                               ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
       IMPLICIT REAL*8(A-H,P-Z)
       INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
       INTEGER*4 IANS
       COMMON /MCOUNT/ MROW,MROW1,MNSEC
       COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
       COMMON /INT4/ MR,MR1,NC,NC1,IC
       DIMENSION NODE(N 1,1),NTSE(1),NTE(1)
       IF(IANS.EQ.2) GOTO 601
       K = 1
       MK = 1
```

```
      NRCTOB = 0
      NRCTOE = 0
      NSTATB = 0
      NSTATE = 0
      C =1.D0
      DO 40 IC = 1,NE1
         K1 = K + MROW1 + MROW + 1
         K2 = K1 - MROW - MK
         K3 = K
         K4 = K + 1
         K5 = K + 2
         K6 = K2 + 1
         K7 = K1 + 2
         K8 = K1 + 1
         NODE(IC,1) = K1
         NODE(IC,2) = K2
         NODE(IC,3) = K3
         NODE(IC,4) = K4
         NODE(IC,5) = K5
         NODE(IC,6) = K6
         NODE(IC,7) = K7
         NODE(IC,8) = K8
         K = K + 2
         MK = MK + 1
         CQ = FLOAT(IC)/FLOAT(MR)
         IF(QC.NE.Q) GOTO 40
         K = K + MP + 2
         MK = 1
         C = C + 1.DC
40    CONTINUE
      WRITE(6,64)
64    FORMAT('1')
      WRITE(6,64)
      IF(NROTO.EQ.0.AND.NSTAT.EQ.0) GOTO 600
      IJK1 = NROTO
      IJK2 = NROTO + MROW - 1
      IJK3 = NSTAT
      IJK4 = NSTAT + MROW - 1
      NRCTOB = NODE(IJK1,3)
      NRCTOE = NODE(IJK2,7)
      NSTATB = NODE(IJK3,3)
      NSTATE = NODE(IJK4,7)
      CONTINUE
      GOTO 60C
601   DO 700 I = 1, NE1
         READ(30,710) NODE(I,1),NODE(I,2),NODE(I,3),NODE(I,4),
     1      NODE(I,5),NODE(I,6),NODE(I,7),NODE(I,8),NTE(I)
710      FORMAT(9I5)
700   CONTINUE
600   RETURN
      END
C
C
C
      SUBROUTINE INIT2(BC,NODE,NBC,PSI,RHOSTA,RHOTT,TTOT,WDOT,PTOT,
     1PRESS,TEMP,VZI,R,CP,G,NSE2,MRP1,N2,NSE,NE1,IANS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++         THIS SUBROUTINE COMPUTES THE INLET CONDITIONS             ++
C ++         THROUGH THE USE OF SUBROUTINE FLOCT.  THE                ++
C ++         SUBROUTINE ALSO DETERMINES THE NODE NUMBERS              ++
C ++         WHERE THE VALUES OF PSI ARE TO BE SPECIFIED.             ++
C ++         THE SUBROUTINE COMPUTES THE INITIAL DISTRIBUTION         ++
C ++         OF THE STREAM FUNCTION FROM THE INLET CONDITIONS         ++
C ++         AND THE BOUNDARY CONDITIONS OF THE PROGRAM TURBC.        ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 MP,MR1,NC,NC1,IC,MRP1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
```

```
      INTEGER*4 IANS
      COMMON /ACCUNT/ MROW,MROW1,MNSEC
      COMMON /ACCUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      DIMENSION RC(1),PSI(1)
      DIMENSION NODE(NE1,1),NBC(1)
      G = 1.4DC
      R = 53.3DC
      GC = 32.174DC
      IF(IANS.EQ.1) GOTO 10
      NNBC = 2 * NCOL1 + MROW1 - 2
      DO 700 I = 1,NNBC
         READ(35,710) NBC(I)
710      FORMAT(I5)
700   CONTINUE
      GOTO 300
10    DO 100 I = 1,MRR1
         NBC(I) = I
100   CONTINUE
      MM = 1
      MK = 1
      K = MRR1 + 1
      JK = K
      L = MR21 + 1
      M = 2 * (NCOL1 - 1) + K
      DO 200 J = K,M
         IF(MM.GT.0) GOTO 210
         NBC(K) = L
         K = K + 1
         L = L + 1
         MM = C - MM
         GOTO 200
210      IF(MK.GT.0) GOTO 220
         NBC(K) = L
         L = L + MROW1 - 1
         MK = C - MK
         K = K + 1
         MM = C - MM
         GOTO 200
220      NBC(K) = L
         L = L + MROW
         MK = C - MK
         K = K + 1
         MM = C - MM
200   CONTINUE
      K = MRR1 + 2 * (NCOL1 - 2) + 1
      JK = N2 - MRR1 + 1
      NBC(K) = J
      NBC(K+1) = N2
300   CONTINUE
      GM1 = G - 1.DO
      GM1I = 1.DC / GM1
      RHUB2 = RC(MROW1) * RC(MROW1)
      AREA = 3.141592654 * ( RC(1) * RC(1) - RHUB2) / 144.0DO
      CALL FLOFCT(WDOT,PTOT,TTOT,RHOTT,VTOT,XVEL,CP,R,GM1,GM1I,
     1AREA)
      QUANT = 1.CDC - XVEL * XVEL
      TEMP = TTOT * QUANT
      PRESS = PTOT * QUANT**(G*GM1I)
      RHOSTA = RHOTT * QUANT**GM1I
      VZI = VTOT * XVEL
      NC = MROW1 - 1
      PSI1 = WDOT / (2.CDO * 3.141592654)
      PSID = PSI1 / FLOAT(NC)
      J = 0
      DO 500 I = 1,NC
         PSI(I) = PSI1 - FLOAT(J) * PSID
         J = J + 1
500   CONTINUE
      PSI(MROW1) = 0.CCCOCCOOOOOOOOO
      DO 600 I = 1,NE1
         NI1 = NODE(I,1)
```

```
          NI2 = NODE(I,2)
          NI3 = NODE(I,3)
          NI4 = NODE(I,4)
          NI5 = NODE(I,5)
          NI6 = NODE(I,6)
          NI7 = NODE(I,7)
          NI8 = NODE(I,8)
          PSI(NI1) = PSI(NI3)
          PSI(NI2) = PSI(NI3)
          PSI(NI8) = PSI(NI4)
          PSI(NI6) = PSI(NI5)
          PSI(NI7) = PSI(NI5)
600   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ERR1(NEXCR)
      NEXCR = -NEXCR
      WRITE(6,100) NEXCR
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR REAL*8 VARIABLES BY
     1',I5//)
      STOP
      END
C
C
C
      SUBROUTINE ERR2(NEXCI)
      NEXCI = -NEXCI
      WRITE(6,100) NEXCI
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR INT*2 VARIABLES BY
     1',I5//)
      STOP
      END
C
C
C
      SUBROUTINE ERR3(NEXC4)
      NEXC4 = -NEXC4
      WRITE(6,100) NEXC4
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR INT*4 VARIABLES BY
     1',I5//)
      STOP
      END
C
C
C
      SUBROUTINE TASK1(RC,R,NSE2,MRW1,N2,NROTOB,NROTOE,NSTATB,NSTATE)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++          THIS SUBROUTINE COMPUTES THE TANGENTIAL BLOCKAGE         ++
C ++          FACTORS FOR THE ROTOR AND STATOR OF THE NASA TASK1      ++
C ++          COMPRESSOR.                                             ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRW1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      COMMON /NCOUNT/ MROW,MROW1,MNSEC
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      DIMENSION RC(1),R(1)
      IF(IANS.EQ.2) GOTO 700
      MM1 = NROTOB + MROW1 - 1
      MM2 = MM1 + 1
      MM3 = MM2 + MROW
      MM4 = MM3 + 1
      MM5 = NSTATB + MROW1 - 1
      MM6 = MM5 + 1
```

```
      MM7 = MM6 + MROW
      MM8 = MM7 + 1
      DO 100 I = 1,NN
        B(I) = 1.0D0
100   CONTINUE
      DO 200 I = NFCTCE,MM1
      RCI = RC(I)
      RTCL = 0.0131143 - 2.49672D-4 * RCI - 1.03091D-5 * RCI * RCI +
     14.24339D-10 * RCI * RCI * RCI * RCI * RCI
      RSIGL = 7.04973 + 1.22251D-3 *RCI*RCI + 1.46066D-8 * RCI * RCI *
     1 RCI * RCI * RCI - 1.33865D-3 * DCOS(RCI) - 7.42494D-4 *DSIN(RCI)
     2 + 2.899621D-5 * DTAN(RCI) - 2.12971 * DLOG(RCI)
      B(I) = 1.D0 - RTCL * RSIGL
200   CONTINUE
      DO 300 I = MM2,MM3
      RCI = RC(I)
      RTCM = 0.140493 - 5.89061D-3 * RCI - 3.28044D-11 *RCI* RCI*RCI
     1 * RCI * RCI * RCI + 2.15909D-15 * RCI * RCI * RCI * RCI * RCI *
     2 RCI * RCI * RCI * RCI - 4.55276D-4 * DCOS(RCI) - 1.2162D-4 *
     3 DSIN(RCI) - 3.84042D-6 * DTAN(RCI)
      RCI = RC(I)
      RSIGM = 10.9594 + .255785 * RCI - 2.62070D-6 * RCI * RCI * RCI *
     1 RCI + 4.45133D-3 * DSIN(RCI) - 1.08378D-4 * DTAN(RCI)
     2 - 4.83569 * DLOG(RCI)
      B(I) = 1.D0 - RTCM * RSIGM
300   CONTINUE
      DO 400 I = MM4,NFCTCE
      RCI = RC(I)
      RTCT = 0.0199736 - 1.11132D-3 * RCI + 1.34790-6 * RCI*RCI*RCI -
     1 1.06339D-14 * RCI * RCI * RCI * RCI * RCI * RCI * RCI * RCI *RCI
     2 + 8.41526D-5 * DCOS(RCI) + 2.68422D-5*DSIN(RCI) - 5.72139D-7 *
     3 DTAN(RCI)
      RSIGT = 17.9259 + .656044 * RCI - 3.96891D-6 * RCI * RCI * RCI *
     1 RCI + 1.65560D-2 * DSIN(RCI) - 1.36477D-4 * DTAN(RCI)
     2 - 9.50809 * DLOG(RCI)
      B(I) = 1.D0 - RTCT * RSIGT
400   CONTINUE
      DO 500 I = NSTATB, MM5
      RCI = RC(I)
      STMCL = 0.0188089 + 2.31157D-3 * RCI + 1.60096D-5 * RCI * RCI -
     17.69543D-5 * DCOS(RCI) + 4.83218D-5 * DSIN(RCI) + 3.84003D-7 *
     2 DTAN(RCI)
      SSIGL = 4.77577 - 0.350357 * RCI + 9.46492D-3 * RCI * RCI -
     1 5.52663D-13 * RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI
     2 - 1.31164D-3 * DSIN(RCI) - 1.72759D-6 * DTAN(RCI)
      STETML = 0.272802 - 9.02555D-3 * RCI + 1.99430-11 *
     1 RCI*RCI*RCI*RCI*RCI*RCI*RCI + 3.05063D-4 * DCOS(RCI)
     2 + 2.25621D-4 * DSIN(RCI) - 0.87345D-6 * DTAN(RCI)
      B(I) = 1.0D0 - STETML * STMCL * SSIGL
500   CONTINUE
      DO 600 I = MM6, MM7
      RCI = RC(I)
      STMCM = 0.0158763 + 2.58621D-3*RCI + 9.85191D-6 * RCI * RCI  -
     1 7.56625D-5 * DCOS(RCI) + 6.89499D-5 * DSIN(RCI) - 4.16541D-6 *
     2 DTAN(RCI)
      SSIGM = 5.17002 - .401549 * RCI + 1.13424D-2 * RCI * RCI -
     1 6.59065D-13 * RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI +
     2 2.91139D-4 * DTAN(RCI)
      B(I) = 1.0D0 - STMCM * SSIGM
600   CONTINUE
      DO 700 I = MM8, NSTATE
      RCI = RC(I)
      STMCT = 0.00994821 + 3.14858D-3 * RCI - 3.60047D-11 * RCI*RCI*RCI
     1 *RCI*RCI*RCI + 1.80029D-5 * DCOS(RCI) - 3.03128D-5 * DSIN(RCI) -
     2 1.39126D-5 * DTAN(RCI)
      SSIGT = 5.90125 - .506055 * RCI + 1.52284D-2 * RCI * RCI -
     1 2.16170D-11 * RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI
     2 + 1.07797D-4 * DTAN(RCI)
      STETMT = 0.615185 + 2.5256D-4 * RCI * RCI - 6.349-14 * RCI *
     1 RCI*RCI*RCI*RCI*RCI*RCI*RCI*RCI + 1.64098D-3 * DCOS(RCI)
     2 - 0.194717 * DLOG(RCI)
      B(I) = 1.0D0 - STETMT * STMCT * SSIGT
```

```
700   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FILGEN(ZC,RC,B,PSI,NODE,NTE,NBC,PRESS,PTOT,TEMP,TTOT,
     1RHOSTA,RHOTT,VZI,WDOT,RPM,CP,R,G,NROTOB,NSTATB,
     2NSE2,NSE,N2,MRR1,NE1,IANS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++    THIS SUBROUTINE WRITES THE OUTPUT OF THE PROGRAM               ++
C ++    ONTO DISK STORAGE.  THE FILE DEFINITIONS ARE LISTED           ++
C ++    IN THE EXEC FILES TURBO1 AND TURBO1A.                         ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      INTEGER*4 IANS
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /NCOUNT/ MROW,MROW1
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
      DIMENSION ZC(1),RC(1),B(1),PSI(1)
      DIMENSION NODE(NE1,1),NTE(1),NBC(1)
C
      IF(IANS.EQ.2) GOTO 101
      DO 100 I=1,NN
         WRITE(20,110) ZC(I),RC(I),B(I)
110      FORMAT(3F15.11)
100   CONTINUE
      WRITE(25,120) NN,MROW,NCOL,MROW1,NCOL1,NROTOB,NSTATB
101   WRITE(25,121)PRESS,PTOT,TEMP,TTOT,RHOSTA,RHOTT,WDOT,CP,R,G,RPM,VZI
120   FORMAT(7I5)
121   FORMAT(4F11.6,2F12.8,/,4F12.7,F8.1,F13.8)
      IF(IANS.EQ.2) GOTO 301
      DO 200 I = 1, NE1
         WRITE(30,210) NODE(I,1),NODE(I,2),NODE(I,3),NODE(I,4),
     1         NODE(I,5),NODE(I,6),NODE(I,7),NODE(I,8),NTE(I)
210      FORMAT(9I5)
200   CONTINUE
      L = 2 * NCOL1 + 2 * MROW1 - 4
      DO 300 I = 1,L
         WRITE(35,310) NBC(I)
310      FORMAT(I5)
300   CONTINUE
301   DO 400 I=1,NN
         WRITE(40,410) PSI(I)
410      FORMAT(F15.11)
400   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE FLOFCT(WDOT,PTOT,TTOT,RHOTT,VTOT,XVEL,CP,R,GM1,GMII,
     1AREA)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++       THIS COMPUTES THE NONDIMENSIONAL VELOCITY X                ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-G,P-Z), REAL*4(H)
      INTEGER*4 MR,MR1,NC,NC1,IC,MRR1,NE1,MN1,MN2,MN3,MN4,NSE2,N2
      COMMON /INT4/ MR,MR1,NC,NC1,IC
      COMMON /NCOUNT/ MROW,MROW1,MNSEC
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NROTC,NSTAT
```

```
      IT = 0
      XVEL = 0.1D0
      EPS = 1.0D-06
      RJ = 778.2 * 32.174
      RHCTT = (PIDT * 144.0D0)/(R * TTOT)
      VTOT = DSQRT(2.0D0 * CP * RJ * TTOT)
      PHI1 = WDOT / (RHCTT * VTOT * AREA)
100   PHI = XVEL * (1.0D0 - XVEL * XVEL)**GM1I
      DIFF = DABS(PHI1 - PHI)
      IF(DIFF.LT.EPS) GOTO 200
      DPHIDX = PHI*(1.D0/XVEL - (2.D0*XVEL)/(GM1*(1.0D0 - XVEL*XVEL)))
      XVEL = XVEL + (PHI1 - PHI) / DPHIDX
      IT = IT + 1
      IF(IT.LT.21) GOTO 100
      WRITE(6,110)
110   FORMAT(' CONVERGENCE NOT REACHED',/)
200   CONTINUE
      VZI = XVEL * VTOT
      RETURN
      END
C
C
C
      SUBROUTINE MPLOT(ZC,RC,ZMAX,ZMIN,RMAX,RMIN,DHZC,DHRC,            MPLO00
     1DVZC,DVRC,CHZC1,CHRC1,CZC,CRC,KLK,NSE2,ARR1,N2,NSE,
     2NE1,NC2,MM2)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                   ++
C ++       THIS SUBROUTINE CREATES A TEKTRONIX 618 PLOT OF             ++
C ++       EIGHT-NODE ISOPARAMETRIC ELEMENT MESH.                      ++
C ++                                                                   ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z)    , REAL*4(C)
      INTEGER*4 MR,MR1,NC,NC1,IC,NE1,N14,MR14,NC14,NCC1
      LOGICAL*1 TITL1
      COMMON /FPAR/ F8(5000)
      COMMON /NPAR/ I2(2000)
      COMMON /CPAR4/ C4(3000)
      COMMON /INT4/ MR,MR1,NC,IC
      COMMON /MCOUNT/ MROW,MROW1,MNSEC,LIM1,LIMR,LIM4
      COMMON /NCOUNT/ NN,NCOL,NCOL1,NE,NPDTC,NSTAT
      DIMENSION CHZC(1),CHRC(1),DVZC(1),DVRC(1),CXYL(4)
      DIMENSION CHZC1(1),CHRC1(1),CZC(1),CRC(1),DPS(10)
      DIMENSION ZC(1),RC(1)
      DIMENSION KLK(1)
      DIMENSION TITL1(37)
      DATA TITL1/'EIGHT NODE ISOPARAMETRIC ELEMENT MESH'/
C ***********************************************************************
C **                                                                   **
C **              CONVERT THE NODAL COORDINATES TO REAL*4              **
C **              TO MAKE THEM COMPATABLE WITH PLOTG CALLS             **
C **                                                                   **
C ***********************************************************************
      DO 30 I = 1,NN
         CZC(I) = ZC(I)
         CRC(I) = RC(I)
30    CONTINUE
      CXYL(1) = ZMIN
      CXYL(2) = RMIN
      CXYL(3) = ZMAX
      CXYL(4) = RMAX
      CALL DSINIT
      CALL GSFFSE
C ***********************************************************************
C **                       PLOT NODAL POINTS                           **
C ***********************************************************************
      DO 5 I = 1,NCOL1
         CHZC(I) = 0.0
```

```
         OHRC(I) = 0.0
5     CONTINUE
      DO 6 I = 1,MROW1
         CVZC(I) = 0.0
         CVRC(I) = 0.0
6     CONTINUE
      MM1 = MROW1 + 1
      NCC1 = NCOL1 + 1
      N14 = NN
      MR14 = MROW1
      NC14 = NCOL1
      NN1 = NN + 1
      MR1 = MROW + 1
      CALL PLOT('MMNNNNNONL',N14,CZC,CRC,OXYL,37,TITL1)
C *************************************************************************
C **               PLOT HORIZONTAL ELEMENT BOUNDARIES                    **
C *************************************************************************
      MM = 1
      KKK = 1
         DO 100 J = 1, NCOL1
            OHZC(J) = CZC(MM)
            OHRC(J) = CRC(MM)
            KLK(J) = MM
            IF(KKK.LT.0) GOTO 90
            MM = MM + MROW1
            KKK = 0 - KKK
            GOTO 100
90          MM = MM + MR1
            KKK = 0 - KKK
100      CONTINUE
         CALL PLOT('MMNNNNNONL',NC14,OHZC,OHRC,OXYL,37,TITL1)
         DO 200 I = 1,MROW
            MLK = 2 * I
            KKK = 1
            DO 800 II = 1,NCOL1
               IF(KKK.LT.0) GOTO 850
               III = KLK(II) + MLK
               CHZC1(II) = CZC(III)
               CHRC1(II) = CRC(III)
               KKK = 0 - KKK
               GOTO 800
850            III = KLK(II) + I
               CHZC1(II) = CZC(III)
               CHRC1(II) = CRC(III)
               KKK = 0 - KKK
800         CONTINUE
C           MM = (2 * I) + 1
C           MLK = MLK + 1
            CALL PLOT('MMNNNNNONL',NC14,CHZC1,CHRC1,OXYL,37,TITL1)
200   CONTINUE
C *************************************************************************
C **                          PLOT VERTICAL BOUNDARIES                   **
C *************************************************************************
      MM = 1
      NC11 = NCOL + 1
      DO 400 I = 1,NC11
         DO 300 J = 1,MROW1
            OVZC(J) = ZC(MM)
            OVRC(J) = RC(MM)
            MM = MM + 1
300      CONTINUE
         CALL PLOT('MMNNNNNONL',MR14,CVZC,CVRC,OXYL,37,TITL1)
         MM = MM + MR1
400   CONTINUE
      CALL DSTERM
      RETURN
      END
```

APPENDIX F

LISTING OF THE PROGRAM TURBO

```
FILE: TURBO     FORTRAN  A1   NAVAL POSTGRADUATE SCHOOL

C *********************************************************************
C *********************************************************************
C *********************************************************************
C *****                                                           *****
C *****                       PROGRAM TURBO                       *****
C *****                                                           *****
C *********************************************************************
C *********************************************************************
C *********************************************************************
C *****                                                           *****
C *****                                                           *****
C *****    THIS MERIDIONAL THROUGH-FLOW ANALYSIS PROGRAM          *****
C *****    APPLIES A GALERKIN FINITE ELEMENT METHOD TO A          *****
C *****    STREAM FUNCTION FORMULATION. THE PROGRAM USES          *****
C *****    EIGHT NODE, ISOPARAMETRIC ELEMENTS AND THREE-          *****
C *****    POINT GAUSS-LEGENDRE QUADRATURE NUMERICAL              *****
C *****    INTEGRATION.  SELECTED RESULTS ARE DISPLAYED           *****
C *****                                                           *****
C *****    ON THE TEKTRONIX 618 GRAPHICS TERMINAL.                *****
C *****                                                           *****
C *********************************************************************
C *********************************************************************
C *********************************************************************
C *********************************************************************
C *********************************************************************
C
C
      IMPLICIT REAL*8(A-H,P-Z), REAL*4(O)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
C
C              DIMENSION ALL ARRAY VARIABLES
C
      COMMON /REAL8/ R8(56000)
      COMMON /REAL4/ O4(500)
      COMMON /INT2/ I2(2000)
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /ROTN/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLET,UOUTLET,PSI1,BTU,F2I2,F1I1,GC
      COMMON /MCOUNT/ MSOR,MROW1,KK
      COMMON /LIO/ NREAD,NWRITE
      COMMON /NPOINT/ NP1,NP2,NP3,NP4,NP5,NP6,NP7,NP8,NP9,NP10,NP11,
     1NP12,NP13,NP14,NP15,NP16,NP17,NP18,NP19,NP20,NP21,NP22,NP23,
     2NP24,NP25,NP26,NP27,NP28,NP29,NP30,NPI1,NPI2,NPI3,NPI4,NPI5,
     3NPO1,NPO2,NPO3,NPO4,NPO5,NPO6
      DIMENSION ZA(9),EA(9),W(3),D(3),E(8),SF(8)
      DIMENSION RC8(8),ZC8(8),EM8(8,8),BBB(3)
      DIMENSION N(8),L1(2),M1(2)
      DIMENSION TITLE(10),RJAC(2,2),RIJAC(2,2)
      DATA STOP/'STOP'/
C
C
      CALL RDATA(ZA,EA,W,RELX,KK,LIMI,LIMR,LIM4)
C
C
      CALL INIT1(LIMI,LIMR,TITLE,NROTOR,NSTATR,LIM4,NNOD,NE4,NNE4,
     1NROWS)
C
C
      CALL ZERO1(R8(NP14),R8(NP12),R8(NP15),R8(NP8),R8(NP9)
     2,R8(NP10),R8(NP23),R8(NP2),R8(NP1),R8(NP3),
     3I2(NPI1),I2(NPI2),R8(NP13),R8(NP16),R8(NP17),R8(NP6),R8(NP7)
     4,R8(NP4),R8(NP11),R8(NP5),R8(NP21),R8(NP22),R8(NP2)),
     5EM8,I2(NPI3),I2(NPI4),R8(NP28),R8(NP29),R8(NP19),NNOD,NE4,
     6NNE4,NROWS)
C
C
      CALL INPUT(R8(NP1),R8(NP2),R8(NP3),R8(NP4),R8(NP5),I2(NPI3),
     1I2(NPI4),R8(NP22),R8(NP21),R8(NP16),R8(NP8),R8(NP12),R8(NP13),
     2R8(NP23),R8(NP24),R8(NP25),R8(NP26),R8(NP27),
     3I2(NPI1),I2(NPI2),TITLE,NROTOR,NNOD,NE4,NNE4,NROWS)
C
```

```
C
C
C
C           CALCULATE 'V' AND 'U' VELOCITIES AND NEW NODAL DENSITY
C
300   CALL DIST(R8(NP2),I2(NPI4),R8(NP1),R8(NP12),R8(NP16),R8(NP17),
     1R8(NP3),R8(NP8),R8(NP9),R8(NP4),R8(NP5),R8(NP6),R8(NP18),R8(NP27)
     2,R8(NP23),R8(NP24),R8(NP26),R8(NP25),R8(NP7),R8(NP10),R8(NP11),
     3I2(NPI3),R8(NP28),R8(NP29),R8(NP21),R8(NP22),R8(NP19),NROTOR,
     4NSTATB,NNOD,NE4,NNE4,NROWS)
C
C
      CALL FCAL(R8(NP14),W,R8(NP6),ZA,EA,R8(NP8),
     1R8(NP2),R8(NP1),R8(NP18),R8(NP10),I2(NPI2),I2(NPI4),R8(NP11),
     2I2(NPI3),R8(NP28),R8(NP29),R8(NP24),NNOD,NE4,NNE4,NROWS)
C
C
      CALL STIFF(R8(NP2),R8(NP1),R8(NP3),EM1,ZA
     1,EA,W,I2(NPI4),R8(NP16),R8(NP12),
     2R8(NP14),R8(NP15),R8(NP20),I2(NPI2),NNOD,NE4,NNE4,NROWS)
C
C
C             SOLVE SYSTEM OF EQUATIONS
C
      CALL DSIMC(R8(NP20),R8(NP14),NNOD,KS)
C
C
C
      CALL REPLA(R8(NP12),R8(NP14),R8(NP15),NNOD,NE4,NNE4)
C
C
C    COMPARE NEW AND OLD STREAM FUNCTION DISTRIBUTIONS
C
      CALL TEST(R8(NP12),R8(NP13),X,NNOD,NE4,NNE4)
C
C
      CALL RELAX(RELX,R8(NP12),R8(NP13),I2(NPI2),NNOD,NE4,NNE4)
C
C              TEST FOR STREAM FUNCTION CONVERGENCE
C
      KK = KK + 1
      IF(X.LE.2.0D-02 ) GOTO 800
C
C
      CALL NODCON(X,R8(NP14),R8(NP12),R8(NP13),R8(NP20),IFL,NNOD,NE4,
     1NNE4)
C
C               NEXT ITERATION
C
      IF(IFL.EQ.0) GOTO 900
      GOTO 300
800   IFL = 1
C
C
C
900   CALL OUTPLT(X,KK,R8(NP8),R8(NP9),R8(NP10),R8(NP11)
     1,R8(NP12),R8(NP2),R8(NP1),R8(NP14),R8(NP18),R8(NP6)
     2,R8(NP7),R8(NP23),R8(NP24),R8(NP25),R8(NP26),R8(NP27),
     3R8(NP4),R8(NP5),NNOD,NE4,NNE4,NROWS)
C
      RMAX = 20.0D0
      RMIN = 0.0D0
C
      CALL MPLOT(R8(NP2),RMAX,RMIN,R8(NP3),C4(NP01),C4(NP02),C4(NP03),   MPL00
     1C4(NP04),NROTOR,R8(NP4),R8(NP5),R8(NP21),R8(NP22),R8(NP19),
     2C4(NP05),NNOD,NE4,NNE4,NROWS)
C
C
      STOP
      END
C
C
```

```
C
C ******************************************************************
C ******************************************************************
C ******************************************************************
C *****                                                        *****
C *****          BEGINING OF THE SUBROUTINE SECTION:          *****
C *****          CONTAINS FCAL, VEL, SLINE, JACOB,            *****
C *****          SHAPE,MPLOT, AND CSIMG.                      *****
C *****                                                        *****
C ******************************************************************
C ******************************************************************
C ******************************************************************
C
C
      SUBROUTINE INITI(LIMI,LIMP,TITLE,NROTOB,NSTATB,LIM4,NNOD,NE4,
     1NNE4,NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                              ++
C ++                                                              ++
C ++                                                              ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMP,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /MCOUNT/ MROW,MROW1,KK
      COMMON /NPOINT/ NP1,NP2,NP3,NP4,NP5,NP6,NP7,NP8,NP9,NP10,NP11,
     1NP12,NP13,NP14,NP15,NP16,NP17,NP18,NP19,NP20,NP21,NP22,NP23,
     2NP24,NP25,NP26,NP27,NP28,NP29,NP30,NPI1,NPI2,NPI3,NPI4,NPI5,
     3NPO1,NPO2,NPO3,NPO4,NPO5,NPO6
      COMMON /LIO/ NREAD,NWRITE
      DIMENSION TITLE(10)
C
      READ(NREAD,100)TITLE
100   FORMAT(10A4)
C
C
C
C                READ IN NUMBER OF NODES AND NUMBER OF ELEMENTS
C
      READ(25,200) NN,MROW,NCOL,MROW1,NCOL1,NROTOB,NSTATB
200   FORMAT(7I5)
C
      NNE = 3
      NE = MROW * NCOL
      NNOD = NN
      NE4 = NE
      NNE4 = NNE
      NROWS = 2 * MROW + 1
      NP1 = 1
      NP2 = NP1 + NNOD
      NP3 = NP2 + NNOD
      NP4 = NP3 + NNOD
      NP5 = NP4 + NNOD
      NP6 = NP5 + NNOD
      NP7 = NP6 + NNOD
      NP8 = NP7 + NNOD
      NP9 = NP8 + NNOD
      NP10 = NP9 + NNOD
      NP11 = NP10 + NNOD
      NP12 = NP11 + NNOD
      NP13 = NP12 + NNOD
      NP14 = NP13 + NNOD
      NP15 = NP14 + NNOD
      NP16 = NP15 + NNOD
      NP17 = NP16 + NNOD
      NP18 = NP17 + NNOD
      NP19 = NP18 + NNOD
      NP20 = NP19 + NNOD
```

```
      NP21 = NP20 + NNOD*NNOD
      NP22 = NP21 + NNOD
      NP23 = NP22 + NNOD
      NP24 = NP23 + NNOD
      NP25 = NP24 + NNOD
      NP26 = NP25 + NNOD
      NP27 = NP26 + NNOD
      NP28 = NP27 + NNOD
      NP29 = NP28 + NNOD
      NP30 = NP29 + NNOD
      NPI1 = 1
      NPI2 = NPI1 + NNOD
      NPI3 = NPI2 + NNOD
      NPI4 = NPI3 + NE4
      NPI5 = NPI4 + NE4*NNE4
      NPC1 = 1
      NPC2 = NPC1 + MROW1
      NPC3 = NPC2 + MROW1
      NPC4 = NPC3 + MROW1
      NPC5 = NPC4 + MROW1
      NPC6 = NPC5 + MROW1
      NEXCR = LIMR - NP30
      NEXCI = LIMI - NPI5
      NEXC4 = LIM4 - NPC6
      IF(NEXCR.LT.0) CALL ERR1(NEXCR)
      IF(NEXCI.LT.0) CALL ERR2(NEXCI)
      IF(NEXC4.LT.0) CALL ERR3(NEXC4)
      WRITE(6,210) NEXCR,NEXCI,NEXC4
210   FORMAT(2X,' MEMORY SPACE AVAILABLE :',/,' REAL 8 =',I5,2X,' INTEGE
     1R = ',I5,2X,' REAL 4 =',I5/////)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE JACOB(E1,Z1,D,E,RC$,ZC$,RJAC)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                   ++
C ++           THIS SUBROUTINE COMPUTES THE JACOBIAN MATRIX             ++
C ++           REQUIRED FOR THE COORDINATE TRANSFORMATIONS.            ++
C ++              CALL STATEMENT DEFINITIONS:                          ++
C ++                    E1 = VALUE OF THE ETA INPUT                    ++
C ++                    Z1 = VALUE OF THE XCIES INPUT                  ++
C ++                    D(I) = THE DERIVATIVE OF SF(I) WRT Z           ++
C ++                    E(I) = THE DERIVATIVE OF SF(I) WRT E           ++
C ++                    RC$ = R COORDINATES OF THE ELEMENT'S NODES     ++
C ++                    ZC$ = Z COORDINATES OF THE ELEMENT'S NODES     ++
C ++                    RJAC = 2X2 JACOBIAN MATRIX OUTPUT              ++
C ++                                                                   ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION RJAC(2,2),D(8),E(8),RC$(8),ZC$(8)
      RJAC(1,1) = 0.D0
      RJAC(1,2) = 0.D0
      RJAC(2,1) = 0.D0
      RJAC(2,2) = 0.D0
      D(1) = (E1 + 2.D0*Z1 + 2.D0*Z1*E1 + E1*E1)/4.D0
      D(2) = -(Z1 + Z1*E1)
      D(3) = (-E1 + 2.D0*Z1 + 2.D0*Z1*E1 - E1*E1)/4.D0
      D(4) = (-1.D0 + E1*E1)/2.D0
      D(5) = (E1 + 2.D0*Z1 - 2.D0*Z1*E1 - E1*E1)/4.D0
      D(6) = -Z1 + Z1*E1
      D(7) = (-E1 + 2.D0*Z1 - 2.D0*Z1*E1 + E1*E1)/4.D0
      D(8) = (1.D0 - E1*E1)/2.D0
      E(1) = (Z1 + 2.D0*E1 + Z1*Z1 + 2.D0*Z1*E1)/4.D0
      E(2) = (1.D0 - Z1*Z1)/2.D0
```

```
      E(3) = (-Z1 + 2.D0*E1 + Z1*Z1 - 2.D0*Z1*E1)/4.D0
      E(4) = -E1 + E1*Z1
      E(5) = (Z1 + 2.D0*E1 - Z1*Z1 - 2.D0*Z1*E1)/4.D0
      E(6) = (-1.D0 + Z1*Z1)/2.D0
      E(7) = (-Z1 + 2.D0*E1 - Z1*Z1 + 2.D0*Z1*E1)/4.D0
      E(8) = -(E1 + Z1*E1)
      DO 100 I = 1,8
         RJAC(1,1) = RJAC(1,1) + D(I)*ZCS(I)
         RJAC(1,2) = RJAC(1,2) + D(I)*RCS(I)
         RJAC(2,1) = RJAC(2,1) + E(I)*ZCS(I)
         RJAC(2,2) = RJAC(2,2) + E(I)*RCS(I)
100   CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DIST(RC,NODE,ZC,PSI,RHO,RHON,B,VZ,VR,ALP,BE,H,WRL
     1,RHOTT,TEMP,TTOT,PTOT,PRESS,HS,VU,WU,NTE,HR,ENTROP,DEV1,PRAT,
     2ETA,NROTOR,ISTATE,NNOD,NE4,NNE4,NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++      THIS SUBROUTINE CALCULATES U AND V VELOCITIES AND A NEW      ++
C ++      NODAL DENSITY FROM A KNOWN PSI DISTRIBUTION AT EACH OF      ++
C ++      THE NODES IN THE SYSTEM.                                    ++
C ++        CALL STATEMENT DEFINITIONS:                               ++
C ++             NE = NUMBER OF ELEMENTS IN THE MESH                  ++
C ++             NN = NUMBER OF NODES IN THE MESH                     ++
C ++             RC = ARRAY OF R NODAL COORDINATES                    ++
C ++             NODE = ARRAY OF ELEMENTAL NODE ASSIGNMENTS           ++
C ++             G = RATIO OF SPECIFIC HEATS                          ++
C ++             RG = GAS CONSTANT                                    ++
C ++             TT = INLET TOTAL TEMPERATURE                         ++
C ++             RHOT = INLET TOTAL DENSITY                           ++
C ++             RHON = WORK VECTOR WITH NEW DENSITY DISTRIBUTION     ++
C ++             ZC = ARRAY OF NODAL Z COORDINATES                    ++
C ++             PSI = NODAL STREAM FUNCTION VECTOR                   ++
C ++             RHO = NODAL STATIC DENSITY VECTOR                    ++
C ++             B = NODAL BLOCKAGE FACTOR                            ++
C ++             UINLET = INLET AXIAL VELOCITY                        ++
C ++             VZ = NODAL AXIAL VELOCITY                            ++
C ++             VR = NODAL RADIAL VELOCITY                           ++
C ++             RHOSTA = INLET STATIC DENSITY                        ++
C ++             ALP = NODAL ABSOLUTE FLOW ANGLE ARRAY                ++
C ++             BE = NODAL RELATIVE FLOW ANGLE VECTOR                ++
C ++             H = NODAL TOTAL ENTHALPY VECTOR                      ++
C ++             WG = ROTOR SPEED IN RAD/SEC                          ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNPC,NNNRC
      COMMON /NCOUNT/ NROW,NROW1,KK
      COMMON /FCON/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLET,UOULET,PSII,RTU,R2T2,R1T1,GC
      DIMENSION ZC(1),RC(1),H(1),B(1),PRESS(1),HS(1),WU(1)
      DIMENSION WRL(1),RHOTT(1),TEMP(1),TTOT(1),PTOT(1)
      DIMENSION VZ(1),VR(1),RHO(1),PSI(1),VU(1)
      DIMENSION D(8),E(8),SF(8),RCS(8),ZCS(8),ETA(1)
      DIMENSION E1(8),Z1(8),ALP(1),BE(1),ENTROP(1),HR(1)
      DIMENSION NODE(NE4,1),ISTA(8),NTE(1)
      DIMENSION DR(8),DZ(8),RJAC(2,2),RHON(1),DEV1(1),PRAT(1)
      DATA Z1/1.D0,0.D0,-1.D0,-1.D0,-1.D0,0.D0,1.D0,1.D0/
      DATA E1/1.D0,1.D0,1.D0,0.D0,-1.D0,-1.D0,-1.D0,0.D0/
C
C              BEGIN WITH MID NODE OF FIRST ELEMENT AND THEN CYCLE
C              THROUGH EACH ELEMENT.
C
```

```
      GM1 = G - 1.CDC
      ISTA(1) = 3
      ISTA(2) = 2
      ISTA(3) = 1
      ISTA(4) = 1
      ISTA(5) = 1
      ISTA(6) = 2
      ISTA(7) = 3
      ISTA(8) = 3
      DO 100 I = 1,MROW1
         H(I) = CP * TTOT(I) * BTU * GC * 144.0D0
         HS(I) = CP * TEMP(I) * BTU * GC * 144.0D0
100   CONTINUE
      DO 200 II = 1,NE
         NTE1 = NTE(II)
         IF(NTE1 - 2) 210,220,230
C
C
C
210      DO 240 M = 1,NNE
         ISTA1 = ISTA(M)
         CALL SLINE(RC,PSI,VZ,VR,NODE,VM1,RCI1,RCI2,ALP1,NTE1,RHO1,
     1RHO,ALP,E1,II,ISTA1,M,RHOTT,TEMP,TTOT,PRESS,PTOT,B,P1,T1,TT1,PT1,
     2RHOT1,H1,HS1,B2,DPSIR2,DPSIZ2,H,HS,Z1,VZ1,VR1,ZC,ENTROP,ENT1,
     3NNOD,NE4,NNE4,NROWS)
         IJK = NE - NROW
         IF(II.LE.IJK) GOTO 211
         IF(ISTA1.NE.3) GOTO 211
         DPSIZ2 = 0.0D0
211      CALL DUCT(B2,RCI1,RCI2,T1,P1,TT1,PT1,RHOT1,H1,HS1,
     2RHO1,ALP1,ALP2,DPSIR2,DPSIZ2,ISTA1,T2,TT2,P2,PT2,RHO2,RHOT2,
     3VM1,VM2,NNOD,NE4,NNE4,NROWS)
         IF(ISTA1.EQ.1) GOTO 240
         NIM = NODE(II,M)
         PRESS(NIM) = P2
         PTOT(NIM) = PT2
         TTOT(NIM) = TT2
         TEMP(NIM) = T2
         H(NIM) = H1
         HS(NIM) = HS1
         RHO(NIM) = RHO2
         RHOTT(NIM) = RHOT2
         ALP(NIM) = ALP2
         VZ(NIM) = 144.0D0 * DPSIR2 / (RHO2 * B2 * RCI2 / 12.D0)
         VR(NIM) = -144.0D0 * DPSIZ2 / (RHO2 * B2 * RCI2 / 12.D0)
         VU(NIM) = VM2 * DTAN(ALP2)
         WRL(NIM) = RCI2 * VU(NIM)
         ENTROP(NIM) = ENT1
240      CONTINUE
         GOTO 200
C
C
C
220      DO 250 M = 1,NNE
         ISTA1 = ISTA(M)
         RMAX1 = RC(NRCTOP)
         RMIN1 = RC(NRCTOP + MROW1 - 1)
         CALL SLINE(RC,PSI,VZ,VR,NODE,VM1,RCI1,RCI2,ALP1,NTE1,RHO1,
     1RHO,ALP,E1,II,ISTA1,M,RHOTT,TEMP,TTOT,PRESS,PTOT,B,P1,T1,TT1,PT1,
     2RHOT1,H1,HS1,B2,DPSIR2,DPSIZ2,H,HS,Z1,VZ1,VR1,ZC,ENTROP,ENT1,
     3NNOD,NE4,NNE4,NROWS)
         CALL ROTO(RCI1,VM1,BETA1,BETA2,RMACHR,RMAX1,RMIN1,
     1DEV,RCI2,RHO1,RHO2,T1,TT1,P1,PT1,ALP1,RHOT1,RHOT2,B2,ENT,DPSIR2,
     2DPSIZ2,DFAC,ALP2,P2,PT2,T2,TT2,TOMEG,VM2,RPASS,NNOD,NE4,NNE4,
     3NROWS)
         NIM = NODE(II,M)
         RFACT = (1.D0 - RPASS)*(1.D0 - RPASS)
         IF(ISTA1 - 2) 253,251,252
251      PRESS(NIM) = (P2 + P1) / 2.0D0
         PTOT(NIM) = (PT2 + PT1) / 2.0D0
         TTOT(NIM) = (TT2 + TT1) / 2.0D0
         TEMP(NIM) = (T2 + T1) / 2.0D0
```

```
      H(NIM) = CP * TTOT(NIM) * BTU * GC * 144.0D0
      HS(NIM) = CP * TEMP(NIM) * BTU * GC * 144.0D0
      RHO(NIM) = (RHO2 + RHO1) / 2.0D0
      RHOTT(NIM) = (RHOT2 + RHOT1) / 2.0D0
      ALP(NIM) = (ALP2 + ALP1) / 2.0D0
      BE(NIM) = (BETA2 + BETA1) / 2.0D0
      VZ2 = 144.0D0 * DPSIR2 / (RHO2 * B2 * RCI2 / 12.D0)
      VR2 = -144.0D0 * DPSIZ2 / (RHO2 * B2 * RCI2 / 12.D0)
      VZ(NIM) = (VZ1 + VZ2) / 2.0D0
      VR(NIM) = (VR1 + VR2) / 2.0D0
      VU(NIM) = (VM2 + VM1) * DTAN(ALP(NIM)) / 2.0D0
      WU(NIM) = (VM2 + VM1) * DTAN(BE(NIM)) / 2.0D0
      WRL(NIM) = (RCI1+RCI2) * VU(NIM)/2.0D0
      HR(NIM) = H(NIM) - WG*(RCI1 + RCI2) * VU(NIM)/2.0D0
      ENTROP(NIM) = 0.5D0 * ENT
      GOTO 250
252   PRESS(NIM) = P2
      PTOT(NIM) = PT2
      TTOT(NIM) = TT2
      TEMP(NIM) = T2
      DEV1(NIM) = DEV
      H(NIM) = CP * TT2 * BTU * GC * 144.0D0
      HS(NIM) = CP * T2 * BTU * GC * 144.0D0
      RHO(NIM) = RHO2
      RHOTT(NIM) = RHOT2
      ALP(NIM) = ALP2
      BE(NIM) = BETA2
      VZ(NIM) = 144.0D0 * DPSIR2 / (RHO2 * B2 * RCI2 / 12.D0)
      VR(NIM) = -144.0D0 * DPSIZ2 / (RHO2 * B2 * RCI2 / 12.D0)
      VU(NIM) = VM2 * DTAN(ALP2)
      WU(NIM) = VM2 * DTAN(BETA2)
      WRL(NIM) = RCI2 * VU(NIM)
      HR(NIM) = H(NIM) - WG * RCI2 * VU(NIM)
      ENTROP(NIM) = ENT
      GOTO 250
253   BE(NIM) = BETA1
      WU(NIM) = VM1 * DTAN(BETA1)
      HR(NIM) = H(NIM) - WG * RCI1 * VU(NIM)
      ETA(NIM) = (TT1/(TT2 - TT1))*(((PT2/PT1)**(GM1/G)) - 1.0D0)
      PRAT(NIM) = PT2/PT1
250   CONTINUE
      GOTO 200
C
C
C
230   DO 260 M = 1,NNE
      ISTA1 = ISTA(M)
      RMAX1 = RC(NSTATB)
      RMIN1 = RC(NSTATB + NROW1 - 1)
      NIM = NODE(I1,M)
      CALL SLINE(RC,PSI,VZ,VR,NODE,VM1,RCI1,RCI2,ALP1,NTB1,RHO1,
     1RHO,ALP,E1,I1,ISTA1,M,RHOTT,TEMP,TTOT,PRESS,PTOT,B,P1,T1,TT1,PT1,
     2RHOT1,H1,HS1,B2,DPSIR2,DPSIZ2,H,HS,Z1,VZ1,VR1,ZC,ENTROP,ENT1,
     3NNOD,NE4,NNE4,NROWS)
      IF(ISTA1.EQ.1) GOTO 264
      BETA1 = ALP1
      GOTO 265
264   BETA1 = ALP(NIM)
265   CALL STAT(RCI1,VM1,BETA1,BETA2,RMACHR,RMAX1,RMIN1,
     1DEV,RCI2,RHO1,RHO2,T1,TT1,P1,PT1,ALP1,RHOT1,RHOT2,B2,ENT,DPSIR2,
     2DPSIZ2,CFAC,ALP2,P2,PT2,T2,TT2,TOMEG,VM2,RPASS,NNOD,NE4,NNE4,
     3NROWS)
      RFACT = (1.D0 - RPASS)*(1.D0 - RPASS)
      IF(ISTA1 - 2) 263,261,262
261   PRESS(NIM) = (P2 + P1) / 2.0D0
      PTOT(NIM) = (PT2 + PT1) / 2.0D0
      TTOT(NIM) = (TT2 + TT1) / 2.0D0
      TEMP(NIM) = (T2 + T1) / 2.0D0
      H(NIM) = H1
      HS(NIM) = HS1
      RHO(NIM) = (RHO2 + RHO1) / 2.0D0
      RHOTT(NIM) = (RHOT2 + RHOT1) / 2.0D0
```

```
          ALP(NIM) = (BETA2 + BETA1) / 2.0DO
          VZ2 = 144.CDC * DPSIR2 / (RHO2 * B2 * RCI2 / 12.DO)
          VR2 = -144.CDC * DPSIZ2 / (RHO2 * B2 * RCI2 / 12.DO)
          VZ(NIM) = (VZ1 + VZ2) / 2.0DO
          VR(NIM) = (VR1 + VR2) / 2.0DO
          VU(NIM) = (VM2 + VM1) * DTAN(ALP(NIM)) / 2.0DO
          ENTROP(NIM) = 0.500D * ENT + ENT1
          WRL(NIM) = RCI2 * VU(NIM)
          GOTO 260
262       PRESS(NIM) = P2
          PTOT(NIM) = PT2
          TTOT(NIM) = TT2
          TEMP(NIM) = T2
          DEV1(NIM) = DEV
          H(NIM) = H1
          HS(NIM) = HS1
          RHO(NIM) = RHO2
          RHOTT(NIM) = RHOT2
          ALP(NIM) = BETA2
          VZ(NIM) = 144.0DO * DPSIR2 / (RHO2 * B2 * RCI2 / 12.DO)
          VR(NIM) = -144.0DO * DPSIZ2 / (RHO2 * B2 * RCI2 / 12.DO)
          VU(NIM) = VM2 * DTAN(ALP2)
          WRL(NIM) = RCI2 * VU(NIM)
          ENTROP(NIM) = ENT + ENT1
          GOTO 260
263       ETA(NIM) = PT2/PT1
          PRAT(NIM) = PT2/PT1
260       CONTINUE
200   CONTINUE
C
C
      RETURN
      END
C
C
C
      SUBROUTINE SHAPE(E,Z,SF)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                   ++
C ++       THIS SUBROUTINE CALCULATES THE SHAPE FUNCTIONS FOR           ++
C ++       LOCAL NODES READ FROM LEFT TO RIGHT, BOTTOM TO TOP          ++
C ++       IN THE Z-E PLANE.                                           ++
C ++          CALL STATEMENT DEFINITIONS:                              ++
C ++             Z = VALUE OF THE EXCISS INPUT                         ++
C ++             E = VALUE OF THE ETA INPUT                            ++
C ++             SF = SHAPE FUNCTION VECTOR                            ++
C ++                                                                   ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION SF(8)
      SF(1) = (Z*E + Z*Z + E*E + Z*Z*E + Z*E*E - 1.DO)/4.DO
      SF(2) = (1.DO + E - Z*Z - Z*Z*E)/2.DO
      SF(3) = (-Z*E + Z*Z + E*E + Z*Z*E - Z*E*E - 1.DO)/4.DO
      SF(4) = (1.DO - E*E - Z + Z*E*E)/2.DO
      SF(5) = (Z*E + Z*Z + E*E - Z*Z*E - Z*E*E - 1.DO)/4.DO
      SF(6) = (1.DO - E - Z*Z + E*Z*Z)/2.DO
      SF(7) = (-Z*E + Z*Z + E*E - E*Z*Z + Z*E*E - 1.DO)/4.DO
      SF(8) = (1.DO - E*E + Z -Z*E*E)/2.DO
      RETURN
      END
C
C
C
C
C
      SUBROUTINE INPUT(ZC,RC,B,ALP,BE,NTE,NODE,PRAT,DEV1,RHO,
     1UVEL,PSI,PSIC,TEMP,TTOT,PRESS,PTOT,RHOTT,NHS,NBC,TITLE,
     2NRCTCR,NNCD,NE4,NN4,NROWS)
```

```
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNCD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNEC,NNNEC
      COMMON /FCOM/ FG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLET,UOULET,PSIT,BTU,F2I2,FII1,GC
      COMMON /NCOUNT/ MROW,MROW1,KK
      COMMON /LIO/ NREAD,NWRITE
      DIMENSION ZC(1),RC(1),B(1),ALP(1),RE(1),RHO(1),UVEL(1)
      DIMENSION PSI(1),PSIO(1),TITLE(10),DEVI(1),PHAT(1)
      DIMENSION TEMP(1),TTOT(1),PRESS(1),PTOT(1),RHOTT(1)
      DIMENSION NES(1),NBC(1)
      DIMENSION NTE(1),NODE(NE4,1)
      DATA STOP/'STOP'/
C
C
C
C  READ NODE NUMBERS,NODAL COORDINATES (IN INCHES), AND
C  NODAL BLOCKAGE FACTOR. INLET STATION ZC() MUST BE = 0.D0.
C  LAST NODAL ZC(NN) MUST BE AT THE OUTLET STATION.
C
      DO 170 I = 1,NN
         READ(20,1000)ZC(I),RC(I),B(I)
1000     FORMAT(3F15.11)
170   CONTINUE
      DO 171 I = 1,NN
         B(I) = B(I) * 0.91
171   CONTINUE
      IT1 = NROTOB
      IT2 = IT1 + MROW1
      IT3 = IT2 + MROW + 1
      IT4 = IT3 + MROW1
      IT5 = IT4 + MROW + 1
      IT6 = IT5 + MROW1
      IT7 = IT6 + MROW + 1
      IT1A = IT1 + 1
      IT2A = IT2 + 1
      IT3A = IT3 + 1
      IT4A = IT4 + 1
      IT5A = IT5 + 1
      IT6A = IT6 + 1
      IT7A = IT7 + 1
      IT3B = IT3A + 1
C      TIP BLOCKAGE FOR 70% DESIGN
C       B(IT3) = B(IT3) * 0.9000
C       B(IT4) = B(IT4) * 0.9200
C       B(IT5) = B(IT5) * 0.9700
C       B(IT3A) = B(IT3A) * 0.9200
C       B(IT4A) = B(IT4A) * 0.9800
C       B(IT5A) = B(IT5A) * 0.9900
C       B(IT3B) = B(IT3B) * 0.9700
C      TIP BLOCKAGE FOR 50% DESIGN
C       B(IT3) = B(IT3) * 0.8500
C       B(IT4) = B(IT4) * 0.8700
C       B(IT5) = B(IT5) * 0.9200
C       B(IT3A) = B(IT3A) * 0.8700
C       B(IT4A) = B(IT4A) * 0.9300
C       B(IT5A) = B(IT5A) * 0.9500
C       B(IT3B) = B(IT3B) * 0.9200
C
C     READ IN CONNECTIVITY MATRIX. LOCAL NODE NUMBERS START
C     AT ELEMENT'S UPPER RIGHT HAND CORNER AND TRAVERSE CCW.
C     READ IN THE ELEMENT TYPE INDENTIFICATION
C
      DO 180 J = 1,NE
         READ(30,1010) NODE(J,1),NODE(J,2),NODE(J,3),NODE(J,4)
```

```
     1    ,NODE(J,5),NODE(J,6),NODE(J,7),NODE(J,8),NTE(J)
1010      FORMAT (9I5)
180   CONTINUE
C      READ IN INLET THERMODYNAMIC QUANTITIES; FLOW RATE,INLET
C      U VELOCITY,OUTLET U VELOCITY,RHOT,RHOSTA,PTOT,TTOT,N
C      UNITS ARE AS FOLLOWS:  FLOW RATE (LBM/SEC)
C      VELOCITY (FT/SEC) ; RHOT AND RHOSTA (LBM/CU FT) ;
C      PTOT (PSF) : TT ( DEGREES RANKINE) : SPEED (RPM)
C
1034  READ(25,1035)PRES,PT,TEM,TT,RHOSTA,RHOT
1035  FORMAT(4F11.6,2F12.8)
C
C  READ IN FLUID/GAS CONSTANTS,R(GAS CONSTANT),GAMMA, CP(BTU/LBM-R)
C
      READ(25,1037)WDOT,CP,RG,G,SPEED,UINLET
1037  FORMAT(4F12.7,F8.1,F13.8)
C
C                FIND OMEGA (RAD/SEC)
C
      WO = SPEED*2.D0*3.14159300/60.D0
      UINLET = UINLET*12.0D0
C
C          COMPUTE THE FIRST ESTIMATES OF UVEL(I) AND RHO(I).
C
      DO 166 I = 1,NN
         UVEL(I) = UINLET
         RHO(I) = RHOSTA
         TEMP(I) = TEM
         TTOT(I) = TT
         PRESS(I) = PRES
         PTOT(I) = PT
         RHOTT(I) = RHOT
166   CONTINUE
C
C                READ NODES WHERE PSI IS SPECIFIED
C
      NNBC = 2 * NCOL1 + NROW1 - 2
      DO 190 I = 1,NNBC
         READ(35,1020) NBC(I)
1020     FORMAT(I5)
190   CONTINUE
C
C  READ IN THE FIRST ESTIMATE OF SYSTEM'S PSI DISTRIBUTION
C
      DO 193 I = 1,NN
         READ(40,1021) PSI(I)
         PSIO(I) = PSI(I)
1021     FORMAT(F15.11)
193   CONTINUE
      UINLT = UINLET / 12.0D0
C
C                FIND NODES WHERE F(R,Z) IS SPECIFIED
      NNNBC = NN - NNBC
C
C
C                PRINT ALL INPUT DATA
C
C
      GO TO 1101
221   WRITE(NWRITE,1038)TITLE
1038  FORMAT(//////////' ',20X,10A4)
      WRITE(NWRITE,1040)NN,NE
1040  FORMAT(' ',5X,'NO. OF NODES = ',I3,27X,
     1'NO. OF ELEMENTS = ',I2,/)
      WRITE(NWRITE,1041)NROW,NCOL
1041  FORMAT(5X,'NO. OF ROWS = ',I3,28X,
     1'NO. OF COLUMNS = ',I2,/)
      WRITE(NWRITE,1045)
1045  FORMAT(' ',20X,'SUMMARY OF NODAL COORDINATES')
C     GOTO 9999
      WRITE(NWRITE,1050)
1050  FORMAT(' ','NODE',5X,'Z(I)',11X,'R(I)',11X,'B(I)',
     1 7X,'ABS FLOW ANG',3X,'REL FLOW ANG' /)
```

```
      C = 1.
      DO 1052 I = 1,NN
         WRITE(NWRITE,1060)(I,ZC(I),RC(I),B(I),ALP(I),BE(I))
         IF(I.EQ.NN) GOTO 1054
         IC = I
         QQ = FLOAT(IC) / 40.
         IF(QQ.NE.C) GOTO 1052
         WRITE(NWRITE,1053)
1053     FORMAT('1')
         WRITE(NWRITE,1045)
         WRITE(NWRITE,1050)
         C = C + 1.
         GOTO 1052
1054     WRITE(NWRITE,1053)
1052  CONTINUE
1060  FORMAT(' ',I3,2X,E13.6,2X,E13.6,2X,E13.6,2X,E13.6,2X,E13.6)
      WRITE(NWRITE,1062)
1062  FORMAT(' ',27X,'SYSTEM TOPOLOGY',//,2X,'ELEMENT',
     110X,'NODES',40X,'TYPE OF ELEMENT')
      WRITE(NWRITE,1063)(I,NODE(I,1),NODE(I,2),NODE(I,3),NODE(I,4)
     1,NODE(I,5),NODE(I,6),NODE(I,7),NODE(I,8),NTE(I),I=1,NE)
1063  FORMAT(' ',3X,I3,6X,I3,3X,I3,3X,I3,3X,I3,3X,I3,3X,I3,3X,I3,3X,I3,
     15X,I3)
      WRITE(NWRITE,1064)WDOT,RHOT,PT,TT, SPEED,UINLT,RG,G,CP
1064  FORMAT(///' ',14X,'INLET THERMODYNAMIC VARIABLES ARE AS FOLLOWS',
     1//,4X,'FLOW RATE = ',E13.6,' LBM/SEC',//,4X,'TOT DENSITY = ',
     2 E13.6,'LBM CU FT',//,4X,'TOT PRESSURE = ',E13.6,' PSI',//
     3,4X,'TOT TEMPERATURE = ',E13.6,' DEG RANKINE',//,
     4 4X,'ROTATIONAL SPEED = ',E13.6,' RPM',//,
     5 4X,'INLET U VELOCITY = ',E13.6,' FT/SEC',//,
     7 4X,'GAS CONSTANT = ',E13.6,//,
     8 4X,'RATIO OF SPECIFIC HEATS = ',E13.6,//,
     9 4X,'SPECIFIC HEAT AT CONSTANT PRESSURE = ',E13.6,/)
      WRITE(NWRITE,1065)RHOSTA
1065  FORMAT(' ',3X,'STATIC DENSITY AT INLET = ',E13.6)
      WRITE(NWRITE,1053)
      WRITE(NWRITE,1070)
1070  FORMAT(' ',4X,'NODES WHERE PSI IS SPECIFIED',///,4X,
     1'NODE NO.',10X,'PSI(I)')
      WRITE(NWRITE,1080)(NBC(I),PSI(NBC(I)),I=1,NNBC)
1080  FORMAT(' ',5X,I3,10X,E13.6)
      WRITE(NWRITE,1053)
9999   RETURN
       END
C
C
C
      SUBROUTINE ZEROI(F,PSI,RHS,UVEL,VVEL,TVEL,PRESS,RC,ZC,
     1B,NFS,NBC,WREL,RHO,RHON,H,HS,ALP,TWEL,BE,DEV1,PRAT,EM,
     2EMS,NTE,NODE,HP,ENTROP,ETA,NNOD,NE4,NNE4,NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                ++
C ++                                                                ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /FCOM/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLT,UOULET,PSI1,RTU,F2T2,F1I1,GC
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /NCOUNT/ MROW,MROW1,KK
      DIMENSION NTE(1),NODE(NE4,1)
      DIMENSION EMS(8,8),F(1),PSI(1),RHS(1),EM(NNOD,1),ETA(1)
      DIMENSION UVEL(1),VVEL(1),TVEL(1),PRESS(1),RC(1),DEV1(1)
      DIMENSION B(1),WREL(1),ZC(1),RHO(1),RHON(1),HP(1),PRAT(1)
      DIMENSION H(1),HS(1),ALP(1),BE(1),TWEL(1),ENTROP(1)
      DIMENSION NFS(1),NBC(1)
C
C     INITIALIZE ALL MATRICES
```

```
      DO 110 I =1,NN
         F(I) = 0.D0
         PSI(I) = 0.D0
         RHS(I) = 0.D0
         UVEL(I) = 0.D0
         VVEL(I) = 0.D0
         TVEL(I) = 0.D0
         PRESS(I) = 0.D0
         RC(I) = 0.D0
         ZC(I) = 0.D0
         B(I) = 0.D0
         NFS(I) = 0
         ETA(I) = 0.0D0
         NBC(I) = 0
         WRL(I) = 0.D0
         RHO(I) = 0.D0
         RHON(I) = 0.D0
         ENTROP(I) = 0.D0
         H(I) = 0.D0
         HS(I) = 0.D0
         HP(I) = 0.D0
         ALP(I) = 0.D0
         TWEL(I) = 0.D0
         BE(I) = 0.D0
         DEVI(I) = 0.0D0
         PRAT(I) = 0.0D0
110   CONTINUE
      DO 150 I = 1,NN
         DO 150 J = 1,NN
            EM(I,J) = 0.D0
150   CONTINUE
      DO 160 I = 1,NNE
         DO 160 J = 1,NNE
            EMS(I,J) = 0.D0
160   CONTINUE
      DO 165 I = 1,NE
         NTE(I) = 0
         DO 165 J = 1,NNE
            NODE(I,J) = 0
165   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE OUTPUT(X,KK,UVEL,VVEL,TVEL,TWEL,PSI,RC,ZC
     1,RHO,WRL,H,HS,TEMP,TTOT,PRESS,PTOT,RHOTT,ALP,BE,
     2NNOD,NE4,NNE4,NROWS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                 ++
C ++                                                                 ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMP,LIMI
      COMMON /NCOUNT/ NCOL,NCOLI,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /LIO/ NREAD,NWRITE
      COMMON /FCOM/ RG,G,CP,PT,TT,WG,WROT,RHOT,RHOSTA,
     1UINLET,UOUTLET,PSII,BTU,F2I2,FI1I,CC
      DIMENSION ZC(1),RC(1),RHO(1),H(1),HS(1),PSI(1)
      DIMENSION UVEL(1),VVEL(1),TVEL(1),WRL(1),TWEL(1)
      DIMENSION TEMP(1),TTOT(1),PRESS(1),PTOT(1),RHOTT(1)
      DIMENSION ALP(1),BE(1)
C
      IF(IFL.EQ.1) GOTO 450
      WRITE(NWRITE,1600)KK,X
1600  FORMAT(' ','PROGRAM TERMINATED ON ITERATION NO.',I3,/,
     1' RESULTS WHICH FOLLOW ARE FOR CONVERGENCE EPSILON = ',D19.12)
C     GOTO 1310
```

```
C       GOTO1104
450     WRITE(NWRITE,1200)KK,X
1200    FORMAT(' ','STREAM FUNCTION CONVERGENCE CRITERION SATISFIED ON ITE
       1RATION  NUMBER ',I3,//,' RESULTS ARE AS FOLLOWS FOR CONVERGENCE
       2EPSILON = ',D19.12)
C                   WRITE OUT RESULTS
C       GOTO 1310
C
C                   CHANGE UNITS OF VELOCITY TO FT/SEC
C
1104    DO 600 I = 1,NN
           UVEL(I) = UVEL(I)/12.D0
           VVEL(I) = VVEL(I) /12.D0
           TVEL(I) = TVEL(I)/12.D0
           TWEL(I) = TWEL(I)/12.D0
           ALP(I) = ALP(I) * 57.29578
           WRL(I) = WRL(I) / 144.D00
           BE(I) = BE(I) * 57.29578
           H(I) = H(I) / (GC*144.0D0*BTU)
           HS(I) = HS(I) / (GC*144.0D0*BTU)
600     CONTINUE
        WRITE(NWRITE,1109)
1109    FORMAT('1')
        WRITE(NWRITE,1110)
1110    FORMAT(' ',/////,37X,'FINITE ELEMENT RESULTS',///,
       1' NODE',5X,'PSI(I)',10X,'VZ',13X,'VR',12X,'R(I)',10X,'DENSITY'/)
        Q = 1.
        DO 1122 I = 1,NN
        WRITE(NWRITE,1120)(I,PSI(I),UVEL(I),VVEL(I),RC(I),RHO(I))
        IF(I.EQ.NN) GOTO 1124
        IC = I
        QQ = FLOAT(IC)/40.
        IF(Q.NE.QQ) GOTO 1123
        Q = Q + 1.
        WRITE(NWRITE,1053)
        WRITE(NWRITE,1110)
        GOTO 1122
1124    WRITE(NWRITE,1053)
1123    CONTINUE
1122    CONTINUE
1120    FORMAT(' ',I3,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6)
1111    WRITE(NWRITE,1121)
1121    FORMAT(' ','NODE',5X,'WRL(I)',10X,'HT',13X,'VT',12X,'WT',14X,'HS')
        Q = 1.
        DO 1322 I = 1,NN
        WRITE(NWRITE,1300)(I,WRL(I),H(I),TVEL(I),TWEL(I),HS(I))
1300    FORMAT(' ',I3,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6)
        IF(I.EQ.NN) GOTO 1324
        IC = I
        QQ = FLOAT(IC)/40.
        IF(Q.NE.QQ) GOTO 1323
        Q = Q + 1.
        WRITE(NWRITE,1053)
        WRITE(NWRITE,1121)
        GOTO 1322
1324    WRITE(NWRITE,1053)
1053    FORMAT('1')
1323    CONTINUE
1322    CONTINUE
        WRITE(NWRITE,1421)
1421    FORMAT(' ',' NODE',5X,'TEMP',11X,'TTOT',11X,'PRESS',11X,
       1'PTOT',11X,'RHOT')
        Q = 1.
        DO 1422 I = 1,NN
        WRITE(NWRITE,1400)(I,TEMP(I),TTOT(I),PRESS(I),PTOT(I),RHOTT(I))
1400    FORMAT(' ',I3,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6,2X,D13.6)
        IF(I.EQ.NN) GOTO 1424
        IC = I
        QQ = FLOAT(IC)/40.
        IF(Q.NE.QQ) GOTO 1423
        Q = Q + 1.
        WRITE(NWRITE,1053)
```

```
      WRITE(NWRITE,1421)
      GOTO 1422
1424  WRITE(NWRITE,1053)
1423  CONTINUE
1422  CONTINUE
         WRITE(NWRITE,110)
110      FORMAT(2X,' I',1X,' ALPHA',3X,' BETA'//)
      DO 100 I = 1,NN
         WRITE(NWRITE,120) I,ALP(I),BE(I)
120      FORMAT(I5,2F10.6)
100   CONTINUE
1310  RETURN
      END
C
C
C
C
C
      SUBROUTINE DUCT(R2,RCI1,RCI2,T1,P1,TT1,PT1,RHOT1,H1,HS1,
     2RHO1,ALP1,ALP2,DPSIR2,DPSIZ2,ISTA1,T2,TT2,P2,PT2,RHO2,RHOT2,
     3VM1,VM2,NNOD,NE4,NNE4,NROWS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                 ++
C ++                                                                 ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /FCOM/ RG,G,CP,PT,TT,WG,WOUT,RHOT,RHOSTA,
     1UINLET,UOULET,PSII,RTU,F2I2,F1I1,GC
      COMMON /NCOUNT/ NROW,NROW1,KK
C
C
C
C
C                 BEGIN WITH MID NODE OF FIRST ELEMENT AND THEN CYCLE
C                 THROUGH EACH ELEMENT.
C
      WRITE(6,556)
      IF(ISTA1.EQ.1) GOTO 300
      IT = 0
      GM1 = G - 1.0D0
      GM1I = 1.0D0 / GM1
      EPS1 = 1.0D-6
      RHO2 = RHO1
      VM2 = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1      (RHO2*R2*RCI2/12.D0)
100   ALP2 = DATAN((RCI1 * VM1 * DTAN(ALP1)) / (RCI2 * VM2))
      T2 = TT1 - GM1 * (VM2*VM2*(1.0D0 + (DTAN(ALP2))**2)) / (2.D0 *
     1 G * RG * GC * 144.D0))
      P2 = PT1 * (T2/TT1)**(G*GM1I)
      RHO2 = (P2 * 144.D0) / (RG * T2)
      VM2N = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1      (RHO2*R2*RCI2/12.D0)
      DIFF1 = DABS(VM2 - VM2N)
      IF(DIFF1.LT.EPS1) GOTO 350
      VM2 = VM2N
      IT = IT + 1
      IF(IT.GT.100) GOTO 200
      GOTO 100
200   WRITE(6,205)
205   FORMAT(' CONVERGENCE NOT REACHED IN 20 ITERATIONS,DUCT'/)
350   CONTINUE
      RMACH2 = VM2N*VM2N*(1.0D0 + (DTAN(ALP2))**2) / (G*RG*GC*T2*144.D)
      QUANT = 1.0D0 + (GM1/2.D0)*RMACH2
      PT2 = PT1
      TT2 = TT1
      RHOT2 = (PT2 * 144.D0) / (RG * TT2)
300   CONTINUE
      RETURN
```

```
      END
C
C
C
C
      SUBROUTINE ROTC(RCI1,VM1,BETA1,BETA2,RMACHR,RMAX1,RMIN1,
     1DEV,RCI2,RHO1,RHO2,T1,TT1,P1,PT1,ALP1,RHOT1,RHOT2,R2,EDT,DPSIR2,
     2DPSIZ2,DFAC,ALP2,P2,PT2,T2,TT2,TOMEG,VM2,RPASS,NNCD,NE4,NNE4,
     3NROWS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                     ++
C ++                                                                     ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNCD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /FCOM/ GC,G,CP,PT,TT,WG,WDOT,RHOT,RHOST1,
     1LINLET,LOUTLET,PSII,BTU,F2I2,FII1,GC
      COMMON /MCOUNT/ MROW,MROW1,KK
C
C              FIND HREL,WRL,AND TWEL AT LOC NODES 3,4,5(ROTOR).
C
      RCI12 = RCI1 * RCI1
      RCI13 = RCI1 * RCI12
      RCI14 = RCI1 * RCI13
      RCI15 = RCI1 * RCI14
      RCI16 = RCI1 * RCI15
      RCI17 = RCI1 * RCI16
      RCI18 = RCI1 * RCI17
      RCI19 = RCI1 * RCI18
      BETA1 = DATAN(WG * RCI1/ VM1) * 57.29578
      BET1 = BETA1 / 57.29578
      W1 = VM1 / DCOS(BET1)
      BCON3 = 2.0D0 * CP * GC * BTU
      TR1 = T1 + W1*W1 / (BCON3 * 144.0D0)
      RMACHR = DSQRT((VM1*VM1*(1.D0 + DTAN(BET1)*DTAN(BET1))) /
     1 (G*GC*RG*TR1*144.D0))
      RKAPPA = 39.6588 + 0.0691161 * RCI12 - 7.89611D-10 * RCI18 +
     1 4.00905D-11 * RCI19 - 0.121315 * DSIN(RCI1) +
     2 3.92005D-4 * DTAN(RCI1)
      TOVC = 0.124881 - 3.59926D-3 * RCI1 - 8.56016D-5 * RCI12
     1 + 5.39194D-12 * RCI17 -7.30391D-7 * DCOS(RCI1) -
     2 9.25533D-5 * DSIN(RCI1)
      SIG = 7.04973 + 1.22751D-3 *RCI12 + 1.46865D-8 * RCI15
     1 - 1.38865D-3 * DCOS(RCI1) - 7.42494D-4 * DSIN(RCI1)
     2 + 2.99621D-5 * DTAN(RCI1) - 2.12-71 * DLOG(RCI1)
      PHI = 258.292 + 2.77161D-4 * RCI14 + 2.2757D-11 * RCI19 -
     1 0.593567 * DCOS(RCI1) - 0.347410 * DSIN(RCI1) +
     2 0.0277612  * DTAN(RCI1) - 98.7006 * DLOG(RCI1)
      TOVC2 = TOVC * TOVC
      TOVC3 = TOVC * TOVC2
      BETA12 = BETA1 * BETA1
      BETA13 = BETA1 * BETA12
      BETA14 = BETA1 * BETA13
      BETA15 = BETA1 * BETA14
      BETA16 = BETA1 * BETA15
      SIG2 = SIG * SIG
      SIG3 = SIG * SIG2
      RPASS = (RMAX1 - RCI1) / (RMAX1 - RMIN1)
      RPASS2 = RPASS * RPASS
      RINCD = BETA1 - RKAPPA
      RKISH = 0.7000
      RKIT = 13.0*TOVC - 76.06*TOVC2 + 189.9*TOVC3
      RITEN = SIG * (0.08*BETA1 - 2.337D-12*BETA16)
      SLOPEN = -0.024*(2.5-SIG) - 0.00224*(1.8 - SIG)*DSQRT(DABS(1.8
     1-SIG))*BETA1-2.149D-8*(26.43-1.0/(SIG*SIG))*BETA13
      RI2D = RKISH * RKIT * RITEN + SLOPEN * PHI
      EXPC1 = 2.7094 - 1.1375*RPASS + 0.4375*RPASS2
```

```
      EXPO2 = ((1.0828 - 0.344*RPASS + 1.563*RPASS2)/RMACHP)**EXPO1
      RIDIFF = -2.8 + 2.55*RPASS + (5.275 + 7.5*RPASS - 2.5 * RPASS2)
     1 * RMACHP**EXPO2
      RIREF = RI2D + RIDIFF
      RKDELS = 0.700
      RKDELT = 4.667*TOVC + 24.45*TOVC2
      DELTEN = (-0.00183 + 0.0257*SIG + 0.000144*SIG2)*BETA1 +
     1 (1.51D-4 - 2.55D-4 * SIG - 3.102D-4 * SIG2)*BETA12    +
     2 (-2.070 + 7.043 * SIG + 3.621 * SIG2)*1.0D-6*BETA13
      SLOPEM = 0.25 + 7.060-4 * BETA1 - 1.286D-5 * BETA12 + 3.108D-7 *
     1 BETA13
      ACON = 3.35 - 0.0124 * BETA1 - 0.0000984 * BETA12
      BCON = 0.0070 - 0.000708 * BETA1 + 1.36D-5 * BETA12
      DDELDI = DEXP(-ACON*SIG) + (BCON/SIG2)*(DSIN(3.141593*SIG /
     2 (1.2*57.29578)))**2
      B = 0.966 - 0.00305 * BETA1 + 6.195D-5 * BETA12 - 1.478D-6 *
     1 BETA13
      DEL2D = RKDELS * RKDELT * DELTEN + (SLOPEM/(SIG**B)) * PHI +
     1 RIDIFF * DDELDI
      ACON2 = -1.75 + 2.5*RPASS + RPASS**6.58
      BCON2 = 0.29 - 5.55*RPASS + 31.84 * RPASS2 - 57.2*RPASS*RPASS2
     1 + 35.15 * RPASS2 * RPASS2
      CCON = 5.43 + 55.6 * (RPASS - 0.535)**2
      DELDIF = ACON2 + BCON2 * RMACHR**CCON
      DELREF = DEL2D + DELDIF
      DEV = DELREF + (RINCD - RIREF)*DDELDI
      BETA2 = BETA1 - PHI - RINCD + DEV
      EPS = 1.0D-6
      IT = 1
      BETA1 = BETA1 / 57.29578
      BETA2 = BETA2 / 57.29578
      RBAR = (RCI1 + RCI2)/2.0D0
      GM1 = G - 1.D0
      GM1I = 1.DC / GM1
      RHO2 = RHO1*0.75
      VM2 = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1       (RHO2*P2*RCI2/12.D0)
600   ALF2 = DATAN((WG*RCI2 - VM2*DTAN(BETA2))/VM2)
      DFAC = 1 - (VM2*DCOS(BETA1))/(VM1*DCOS(BETA2))+((VM1*DTAN(BETA1)*
     1 RCI1 - VM2*DTAN(BETA2)*RCI2)*DCOS(BETA1))/(2.D0*SIG*VM1*RBAR)
      DFAC2 = DFAC*DFAC
      DFAC3 = DFAC2 * DFAC
      DFAC5 = DFAC2 * DFAC3
      TOMEG1 = (0.0021605B + 0.0304237*DFAC - 0.338447*DFAC2 +
     1 0.955990*DFAC3 - 0.602749*DFAC5) * (2.D00*SIG/DCOS(BETA2))
      TOMEG3 = (0.0021205B + 0.0319210*DFAC - 0.109476*DFAC2 +
     1 0.223078*DFAC3 - 0.0433925*DFAC5) * (2.D00*SIG/DCOS(BETA2))
      TDIFF = (TOMEG1 - TOMEG3)
      IF(RPASS.GE.0.30) GOTO 100
      IF(RPASS.LE.0.10) GOTO 110
      TOMEG = TOMEG3 + (TOMEG1 - TOMEG3) * ((0.3-RPASS)/0.20)**2
      GOTO 120
110   TOMEG = TOMEG1
      GOTO 120
100   TOMEG = TOMEG3
120   TE1 = TR1 + WG*WG * (RCI2*RCI2 - RCI1*RCI1) / (BCON3 * 144.D00)
      PE1 = PT1 * (TE1/TT1)**(G*GM1I)
      PR1 = PT1 * (TR1/TT1)**(G*GM1I)
      PE2 = PE1 - TOMEG*(PR1 - P1)
      W2 = VM2 / DCOS(BETA2)
      T2 = TE1 - W2*W2 / (BCON3 * 144.D00)
      P2 = PE2 * (T2 / TE1)**(G*GM1I)
      RHO2 = (P2*144.D0) / (RG * T2)
      VM2N = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1       (RHO2*P2*RCI2/12.D0)
      DTEST = DABS(VM2 - VM2N)
      IT = IT + 1
      IF(DTEST.LT.EPS) GOTO 700
      VM2 = VM2N
      IF(IT.LT.100) GOTO 600
      WRITE(6,650)
650   FORMAT(' CONVERGENCE NOT REACHED IN 20 ITERATIONS'/)
```

```
700   CONTINUE
      RMACH2 = VM2N*VM2N*(1.0D0 + (DTAN(ALP2))**2)/(G*RG*GC*T2*144.0D0)
      QUANT = 1.0D0 + (GM1/2.D0)*RMACH2
      PT2 = P2 * QUANT**(G*GM1I)
      TT2 = T2 * QUANT
      RHOT2 = (PT2 * 144.D0) / (RG * TT2)
      ENT = -RG*DLOG(PT2/PT1) * GC * 144.0D0
      RETURN
      END
C
C
C
C
      SUBROUTINE STAT(RCI1,VM1,BETA1,BETA2,RMACHR,RMAX1,RMIN1,
     1DEV,RCI2,RHO1,RHO2,T1,TT1,P1,PT1,ALP1,RHOT1,RHOT2,B2,ENT,DPSIR2,
     2DPSIZ2,DFAC,ALF2,P2,PT2,T2,TT2,TOMEG,VM2,RPASS,NNCD,NE4,NNE4,
     3NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNCD,NE4,NNE4,NROWS,LIMR,LIMI
       COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NIMBC
       COMMON /FCOM/ RC,C,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1LINLET,LOUTLET,P3I1,BTU,F2I2,F1I1,GC
       COMMON /NCOUNT/ MROW,MROW1,KK
C
C                FIND HREL,WRL,AND TWEL AT LCC NODES 3,4,5(ROTOR).
C
      RCI12 = RCI1 * RCI1
      RCI13 = RCI1 * RCI12
      RCI14 = RCI1 * RCI13
      RCI15 = RCI1 * RCI14
      RCI16 = RCI1 * RCI15
      RCI17 = RCI1 * RCI16
      RCI18 = RCI1 * RCI17
      RCI19 = RCI1 * RCI18
      W1 = VM1 / DCOS(BETA1)
      BCON3 = 2.0D0 * CP * GC * BTU
      RMACHR = DSQRT((VM1*VM1*(1.D0 + DTAN(BETA1)*DTAN(BETA1))) /
     1 (G*GC*RC*T1*144.D0))
      RKAPPA = 64.6522 - 1.89349 * RCI1 + 1.68514D-8 * RCI17
     1 - .174036 * DCOS(RCI1) + .43145 * DSIN(RCI1)
      TOVC = 0.0188099 + 2.31157D-3 * RCI1 + 1.60096D-5 * RCI12 -
     17.69943D-5 * DCOS(RCI1) + 4.83218D-5 * DSIN(RCI1) + 3.04003D-7 *
     2 DTAN(RCI1)
      SIG = 4.77577 - 0.350357 * RCI1 + 9.65492D-3 * RCI12 -
     15.52663D-13*RCI19 - 1.31164D-3*DSIN(RCI1) - 1.72765D-6*DTAN(RCI1)
      PHI = 166.588 + 1.11179D-10 * RCI19 - .925595 * DCOS(RCI1) +
     1 1.52365*DSIN(RCI1) + .023427*DTAN(RCI1) - 45.7749*DLOG(RCI1)
      TOVC2 = TOVC * TOVC
      TOVC3 = TOVC * TOVC2
      BETA1 = BETA1 * 57.29578
      BETA12 = BETA1 * BETA1
      BETA13 = BETA1 * BETA12
      BETA14 = BETA1 * BETA13
      BETA15 = BETA1 * BETA14
      BETA16 = BETA1 * BETA15
      SIG2 = SIG * SIG
      SIG3 = SIG * SIG2
      RPASS = (RMAX1 - RCI1) / (RMAX1 - RMIN1)
      RPASS2 = RPASS * RPASS
      RINCD = BETA1 - RKAPPA
      RKISH = 0.70D0
      RKIT = 13.0*TOVC - 78.06*TOVC2 + 139.8*TOVC3
      RITEN = SIG * (0.03*BETA1 - 2.337D-12*BETA16)
      SLOPEN = -0.024*(2.5-SIG) - 0.002264*(1.8 - SIG)*DSQRT(DABS(1.8
```

```
     1-SIG))*BETA1-2.169D-9*(26.43-1.0/(SIG*SIG))*BETA13
      RI2D = RKISH * RKIT * RITEN + SLOPEN * PHI
      EXPO1 = 2.7094 - 1.1375*RPASS + 0.4375*RPASS2
      EXPO2 = ((1.0828 - 0.344*RPASS + 1.563*RPASS2)/RMACHR)**EXPO1
      RIDIFF = -2.8 + 2.55*RPASS + (5.275 + 7.5*RPASS - 2.5 * RPASS2)
     1 * RMACHR**EXPO2
      RIREF = RI2D + RIDIFF
      RKDELS = 0.70D0
      RKDELT = 4.667*TOVC + 24.45*TOVC2
      DELTEN = (-0.00143 + 0.0257*SIG + 0.000144*SIG2)*BETA1 +
     1 (1.51D-4 - 2.55D-4 * SIG - 3.102D-4 * SIG2)*BETA12    +
     2 (-2.07D + 7.043 * SIG + 3.621 * SIG2)*1.0D-6*BETA13
      SLOPEM = 0.25 + 7.06D-4 * BETA1 - 1.256D-5 * BETA12 + 3.108D-7 *
     1 BETA13
      ACON = 3.35 - 0.0124 * BETA1 - 0.000984 * BETA12
      BCON = 0.0070 - 0.000703 * BETA1 + 1.36D-5 * BETA12
      DDELDI = DEXP(-ACON*SIG) + (BCON/SIG2)*(DSIN(3.141593*SIG /
     2 (1.2*57.29578)))**2
      B = 0.966 - 0.00205 * BETA1 + 6.135D-5 * BETA12 - 1.4738D-6 *
     1 BETA13
      DEL2D = RKDELS * RKDELT * DELTEN + (SLOPEM/(SIG**B)) * PHI +
     1 RIDIFF * DDELDI
      ACON2 = -1.75 + 2.5*RPASS + RPASS**6.58
      BCON2 = 0.29 - 5.55*RPASS + 31.64 * RPASS2 - 57.2*RPASS*RPASS2
     1 + 35.15 * RPASS2 * RPASS2
      CCON = 5.43 + 55.6 * (RPASS - 0.525)**2
      DELDIF = ACON2 + BCON2 * RMACHR**CCON
      DELREF = DEL2D + DELDIF
      DEV = DELREF + (RINCD - RIREF)*DDELDI
      BETA2 = BETA1 - PHI - RINCD + DEV
      RBAR = (RCI1 + RCI2)/2.0D0
      EPS = 1.0D-6
      IT = 1
      BETA1 = BETA1 / 57.29578
      BETA2 = BETA2 / 57.29578
      GM1 = G - 1.0D
      GM1I = 1.D0 / GM1
      RHO2 = RHO1
      TT2 = TT1
      V1 = DSQRT(VM1*VM1*(1.0D0 + DTAN(BETA1)*DTAN(BETA1)))
      VM2 = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1        (RHO2*R2*RCI2/12.D0)
600   V2 = DSQRT(VM2*VM2*(1.0D0 + DTAN(BETA2)*DTAN(BETA2)))
      DFAC = 1 - V2/V1 + (VM1*DTAN(BETA1)*RCI1
     1   - VM2*DTAN(BETA2)*RCI2)/(2.DC*SIG*V1*RBAR)
      TOMEG = (0.0031208 + 0.0219210*DFAC - 0.109476*DFAC2 +
     1 0.223376*DFAC3 - 0.0438925*DFAC5) * (2.D00*SIG/DCOS(BETA2))
      PT2 = PT1 - TOMEG*(PT1 - P1)
      T2 = TT1 - GM1 * (VM2*VM2*(1.DC + (DTAN(ALP2))**2)) / (2.D0 *
     1 G * RG * GC * 144.0D0)
      P2 = PT2 * (T2 / TT1)**(G*GM1I)
      RHO2 = (P2*144.D0) / (RG * T2)
      VM2N = (DSQRT(DPSIR2*DPSIR2+DPSIZ2*DPSIZ2)) * 144.0D0 /
     1        (RHO2*R2*RCI2/12.D0)
      DTEST = DABS(VM2 - VM2N)
      IT = IT + 1
      IF(DTEST.LT.EPS) GOTO 700
      VM2 = VM2N
      IF(IT.LT.100) GOTO 600
      WRITE(6,650)
650   FORMAT(' CONVERGENCE NOT REACHED IN 20 ITERATIONS'/)
700   RMACH2 = VM2N*VM2N*(1.0D0+(DTAN(BETA2))**2)/(G*RG*GC*T2*144.0D0)
      QUANT = 1.0D0 + (GM1/2.D0)*RMACH2
      RHOT2 = (PT2 * 144.D0) / (RG * TT2)
      ENT = -RG*DLOG(PT2/PT1) * GC * 144.0D0
      RETURN
      END
C
C
C
      SUBROUTINE RDATA(ZA,RA,W,RELX,KK,LIM1,LIMR,LIM4)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /FCOM/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLET,UOULET,PSII,BTU,F2I2,F1I1,GC
      COMMON /LIO/ NREAD,NWRITE
      DIMENSION ZA(9),EA(9),W(9)
C
C
      NREAD = 5
      NWRITE = 6
      RELX =  0.2400D0
C
C       INITIALIZE STREAM FUNCTION ITERATION COUNTER ,KK. SET SIZE
C       OR REAL*8 AND INTEGER*2 STORAGE
C
      KK = 0
      LIMR = 56000
      LIMI = 2000
      LIM4 = 500
C
C             THREE-POINT GAUSSIAN ABSCISSAS
C
      ZZ1 = 0.774596669241483
      ZZ2 = 0.000000000000000
      ZA(1) = ZZ1
      ZA(2) = ZZ1
      ZA(3) = ZZ1
      ZA(4) = ZZ2
      ZA(5) = ZZ2
      ZA(6) = ZZ2
      ZA(7) = -ZZ1
      ZA(8) = -ZZ1
      ZA(9) = -ZZ1
      EA(1) = ZZ1
      EA(2) = ZZ2
      EA(3) = -ZZ1
      EA(4) = ZZ1
      EA(5) = ZZ2
      EA(6) = -ZZ1
      EA(7) = ZZ1
      EA(8) = ZZ2
      EA(9) = -ZZ1
C
C          WEIGHTING VALUES FOR 3 PT. GAUSSIAN QUADRATURE
C
      WW1 = 0.55555555555556
      WW2 = 0.88888888888889
      W(1) = WW1*WW1
      W(2) = WW2*WW1
      W(3) = WW1*WW1
      W(4) = WW2*WW1
      W(5) = WW2*WW2
      W(6) = WW2*WW1
      W(7) = WW1*WW1
      W(8) = WW2*WW1
      W(9) = WW1*WW1
C
C     CONSTANTS USED FOR UNITS CONVERSIONS
C
      BTU = 778.2D0
      F1I1 = 12.0D0
      F2I2 = 144.0D0
      F3I3 = F1I1 * F2I2
      GC = 32.174D0
      RETURN
      END
```

```
C
C
C
C
C
C
C
      SUBROUTINE ERR1(NEXCR)
      NEXCR = -NEXCR
      WRITE(6,100) NEXCR
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR REAL*8 VARIABLES BY
     1',I10//)
      STOP
      END
C
C
C
      SUBROUTINE ERR2(NEXCI)
      NEXCI = -NEXCI
      WRITE(6,100) NEXCI
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR INT*2 VARIABLES BY '
     1,I10//)
      STOP
      END
C
C
C
      SUBROUTINE ERR3(NEXC4)
      NEXC4 = -NEXC4
      WRITE(6,100) NEXC4
100   FORMAT(' EXCEEDED MAXIMUM ALLOWABLE SPACE FOR REAL*2 VARIABLES BY
     1',I10//)
      STOP
      END
C
C
C
      SUBROUTINE MPLOT(RC,RMAX,RMIN,VZ,CVEL,CRC,OBE,OALP,NROTOR,ALP,    MPLO004
     1BE,DEV1,PRAT,ETA,NNCD,CRAT,NE4,NNE4,NROWS)                        MPLO004
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                      ++
C ++        THIS SUBROUTINE CREATES A TEKTRONIX 618 PLOT OF                ++
C ++        THE FOLLOWING PARAMETERS:                                     ++
C ++               AXIAL VELOCITY AT THE ROTOR INLET                      ++
C ++               RELATIVE FLOW ANGLES AT THE ROTOR INLET                ++
C ++               AXIAL VELOCITY AT THE ROTOR OUTLET                     ++
C ++               RELATIVE FLOW ANGLES AT THE ROTOR INLET                ++
C ++               ABSOLUTE FLOW ANGLES AT THE ROTOR INLET                ++
C ++               AXIAL VELOCITY AT THE STATOR INLET                     ++
C ++               ABSOLUTE FLOW ANGLES AT THE STATOR INLET               ++
C ++               AXIAL VELOCITY AT THE STATOR OUTLET                    ++
C ++               ABSOLUTE FLOW ANGLES AT THE STATOR OUTLET              ++
C ++               ADIABATIC EFFICIENCY AT THE ROTOR INLET                ++
C ++               RELATIVE DEVIATION ANGLE AT THE ROTOR OUTLET           ++
C ++               ABSOLUTE DEVIATION ANGLE AT THE STATOR OUTLET          ++
C ++               STATIC PRESSURE RATIO FOR THE ROTOR                    ++
C ++               STATIC PRESSURE RATIO FOR THE STATOR                   ++
C ++                                                                      ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z), REAL*4(O)
      INTEGER*4 NREAD,NWRITE,IC,NNCD,NE4,NNE4,NROWS,LIMR,LIMI
      LOGICAL*1 TITL1(27),TITL2(38),TITL3(39),TITL4(39),TITL5(35)
      LOGICAL*1 TITL6(36),TITL7(36),TITL8(36),TITL9(37),TITL10(35)
      LOGICAL*1 TITL11(28),TITL12(29),TITL13(27),TITL14(28)
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /MCOUNT/ MROW,MROW1,KK
      DIMENSION CRC(1),CVEL(1),PXYL(4),OBE(1),OALP(1),ORAT(1)
      DIMENSION RC(1),VZ(1),BE(1),ALP(1),DEV1(1),PRAT(1),ETA(1)
```

```
C ***********************************************************************
C **                                                                   **
C **          PLOT THE ROTOR INLET VELOCITY DISTRIBUTION               **
C **                                                                   **
C ***********************************************************************
      DATA TITL1/'AXIAL VELOCITY PROFILE AT ROTOR INLET'/
      DATA TITL2/'AXIAL VELOCITY PROFILE AT ROTOR OUTLET'/
      DATA TITL3/'AXIAL VELOCITY PROFILE AT STATOR INLET'/
      DATA TITL4/'AXIAL VELOCITY PROFILE AT STATOR OUTLET'/
      DATA TITL5/'RELATIVE FLOW ANGLES AT ROTOR INLET'/
      DATA TITL6/'RELATIVE FLOW ANGLES AT ROTOR OUTLET'/
      DATA TITL7/'ABSOLUTE FLOW ANGLES AT ROTOR OUTLET'/
      DATA TITL8/'ABSOLUTE FLOW ANGLES AT STATOR INLET'/
      DATA TITL9/'ABSOLUTE FLOW ANGLES AT STATOR OUTLET'/
      DATA TITL10/'ADIABATIC EFFICIENCY AT ROTOR INLET'/
      DATA TITL11/'DEVIATION ANGLE ROTOR OUTLET'/
      DATA TITL12/'DEVIATION ANGLE STATOR OUTLET'/
      DATA TITL13/'TOTAL PRESSURE RATIO, ROTOR'/
      DATA TITL14/'TOTAL PRESSURE RATIO, STATOR'/
      WRITE(6,100)
100   FORMAT(5X,' >DO YOU WISH TO ENTER THE PLOT SEQUENCE? ',/,'
     1 1 = YES; 2 = NO. ',/)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      J = NROTO2
      DO 30 I = 1,MROW1
         ORO(I) = RO(J)
         OVEL(I) = VZ(J)
         OBE(I) = BE(J)
         OALP(I) = ETA(J)
         ORAT(I) = PRAT(J)
         J = J + 1
30    CONTINUE
      NR14 = MROW1
      OXYL(1) = -50.0D0
      OXYL(2) = RMIN
      OXYL(3) = 700.0D0
      OXYL(4) = RMAX
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OVEL,ORO,OXYL,37,TITL1)
      CALL PLOT('MMGNNN3OWL',NR14,OVEL,ORO,OXYL,37,TITL1)
      CALL DSTERM
      WRITE(6,110)
110   FORMAT(5X,' >DO YOU WISH TO CONTINUE PLOT SEQUENCE? ',/,'
     1 1 = YES; 2 = NO. ',/)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OBE,ORO,OXYL,35,TITL5)
      CALL PLOT('MMGNNN3OWL',NR14,OBE,ORO,OXYL,35,TITL5)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,ORAT,ORO,OXYL,27,TITL13)
      CALL PLOT('MMGNNN3OWL',NR14,ORAT,ORO,OXYL,27,TITL13)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OALP,ORO,OXYL,35,TITL10)
      CALL PLOT('MMGNNN3OWL',NR14,OALP,ORO,OXYL,35,TITL10)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
```

```
      J = J + NROW + 1
      DO 40 I = 1,NROW1
         ORC(I) = RC(J)
         OVEL(I) = VZ(J)
         OBE(I) = BE(J)
         CALP(I) = DEV1(J)
         J = J + 1
40    CONTINUE
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OVEL,ORC,OXYL,38,TITL2)
      CALL PLOT('MMGNNN3OWL',NR14,OVEL,ORC,OXYL,38,TITL2)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OBE,ORC,OXYL,36,TITL6)
      CALL PLOT('MMGNNN3OWL',NR14,OBE,ORC,OXYL,36,TITL6)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,CALP,ORC,OXYL,28,TITL11)
      CALL PLOT('MMGNNN3OWL',NR14,CALP,ORC,OXYL,28,TITL11)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      J = J - NROW1
      DO 41 I = 1,NROW1
         ORC(I) = RC(J)
         OBE(I) = ALF(J)
         J = J + 1
41    CONTINUE
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OBE,ORC,OXYL,36,TITL7)
      CALL PLOT('MMGNNN3OWL',NR14,OBE,ORC,OXYL,36,TITL7)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      J = J + NROW + 1
      DO 50 I = 1,NROW1
         ORC(I) = RC(J)
         OVEL(I) = VZ(J)
         ORAT(I) = PRAT(J)
         OBE(I) = ALF(J)
         J = J + 1
50    CONTINUE
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OVEL,ORC,OXYL,38,TITL3)
      CALL PLOT('MMGNNN3OWL',NR14,OVEL,ORC,OXYL,38,TITL3)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OBE,ORC,OXYL,36,TITL8)
      CALL PLOT('MMGNNN3OWL',NR14,OBE,ORC,OXYL,36,TITL8)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
```

```
      CALL PLOT('MMGNNN3NWL',NR14,ORAT,ORC,CXYL,28,TITL14)
      CALL PLOT('MMGNNN3OWL',NR14,ORAT,ORC,CXYL,28,TITL14)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      J = J + MROW + 1
      DO 60 I = 1,MROW1
         ORC(I) = RC(J)
         OVEL(I) = VZ(J)
         OBE(I) = ALP(J)
         OALP(I) = DEV1(J)
         J = J + 1
60    CONTINUE
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OVEL,ORC,CXYL,39,TITL4)
      CALL PLOT('MMGNNN3OWL',NR14,OVEL,ORC,CXYL,39,TITL4)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OBE,ORC,CXYL,37,TITL9)
      CALL PLOT('MMGNNN3OWL',NR14,OBE,ORC,CXYL,37,TITL9)
      CALL DSTERM
      WRITE(6,110)
      READ(15,*) NANS
      IF(NANS.EQ.2) GOTO 500
      CALL DSINIT
      CALL GSERSE
      CALL PLOT('MMGNNN3NWL',NR14,OALP,ORC,CXYL,29,TITL12)
      CALL PLOT('MMGNNN3OWL',NR14,OALP,ORC,CXYL,29,TITL12)
      CALL DSTERM
500   CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE SLINE(RC,PSI,VZ,VR,NODE,VM1,RCI1,RCI2,ALP1,NTE1,RHO1,
     1RHO,ALP,E1,II,ISTA1,M,RHOTT,TEMP,TTOT,PRESS,PTOT,R,R1,T1,TT1,PT1,
     2RHOT1,H1,HS1,B2,DPSIR2,DPSIZ2,H,HS,Z1,VZ1,VR1,ZC,ENTROP,ENT1,
     3NNOD,NE4,NNE4,NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                    ++
C ++                                                                    ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /FLOW/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA,
     1UINLET,UOULET,PSII,BTU,F2I2,FII1,GC
      COMMON /NCOUNT/ MROW,MROW1,KK
      DIMENSION RC(1),VR(1),E1(3),SE(8),RHO(1),RHOTT(1),TEMP(1),TTOT(1)
      DIMENSION VZ(1),VU(1),WRL(1),TWEL(1),PRESS(1),PTOT(1),R(1)
      DIMENSION PSI(1),ALP(1),BF(1),H(1),HS(1),Z1(3),P(8),E(8),ZC(1)
      DIMENSION RC8(8),ZC8(8),RJAC(2,2),DZ(8),DR(8),DPSIP(8),DPSIZ(8)
      DIMENSION ENTROP(1)
      DIMENSION INLET(1),NTE(I),NODE(NE4,1)
C
C
C
C
C             BEGIN WITH MID NODE OF FIRST ELEMENT AND THEN CYCLE
C             THROUGH EACH ELEMENT.
```

```
      NI1 = NODE(II,M)
      P = PSI(NI1)
      IT = 0
      K = II
      IF(ISTA1.EQ.1.AND.NTE1.EQ.1) GOTO 700
      IF(ISTA1.EQ.1) GOTO 200
      RCI2 = RC(NI1)
      B2 = B(NI1)
C           CHECK TO SEE IF P IS WITHIN THE PRESENT ELEMENT
130   IF(P.GE.PSI(NODE(K,5)))GOTO140
C           CHECK NEXT ELEMENT BELOW THE PRESENT ELEMENT.
      K = K + 1
      GOTO130
140   IF(P.GT.PSI(NODE(K,4)))GOTO170
      EL = E1(4)
      ER = E1(5)
150   E2 = (EL + ER)/2.D0
      CALL SHAPE(E2,-1.D0),SF)
      PA = SF(3)*PSI(NODE(K,3)) + SF(4)*PSI(NODE(K,4))
     1     + SF(5)*PSI(NODE(K,5))
C           CHECK FOR STREAMLINE CONVERGENCE
      EPS = DABS(PA - P)
      IF(EPS.LT.1.D-06)GOTO190
      IT = IT + 1
      IF(IT.GT.15)GOTO190
      IF(PA.LT.P)GOTO160
      EL = E2
      GOTO150
160   ER = E2
      GOTO150
170   IF(P.GT.PSI(NODE(K,3)))GOTO185
165   EL = E1(3)
      ER = E1(4)
      GOTO150
C           CHECK NEXT ELEMENT ABOVE PRESENT ELEMENT
185   K = K - 1
      GOTO140
C
C           IF CONVERGENCE CRITERIA SATISFIED.
C           CALCULATE WHIRL AND STATIC ENTHALPY
C
C
C
190   NK3 = NODE(K,3)
      NK4 = NODE(K,4)
      NK5 = NODE(K,5)
      RCI1 = SF(3) * RC(NK3) + SF(4) * RC(NK4) +
     1     SF(5) * RC(NK5)
      VZ1 = SF(3) * VZ(NK3) + SF(4) * VZ(NK4) +
     1     SF(5) * VZ(NK5)
      VR1 = SF(3) * VR(NK3) + SF(4) * VR(NK4) +
     1     SF(5) * VR(NK5)
      VM1 = DSQRT(VZ1 * VZ1 + VR1 * VR1 )
      ALP1 = SF(3) * ALP(NK3) + SF(4) * ALP(NK4) +
     1     SF(5) * ALP(NK5)
      RHO1 = SF(3) * RHO(NK3) + SF(4) * RHO(NK4) +
     1     SF(5) * RHO(NK5)
      RHOT1 = SF(3)*RHOTT(NK3) + SF(4)*RHOTT(NK4) +
     1     SF(5) * RHOTT(NK5)
      T1 = SF(3) * TEMP(NK3) + SF(4) * TEMP(NK4) +
     1     SF(5) * TEMP(NK5)
      TT1 = SF(3) * TTOT(NK3) + SF(4) * TTOT(NK4) +
     1     SF(5) * TTOT(NK5)
      PT1 = SF(3) * PTOT(NK3) + SF(4) * PTOT(NK4) +
     1     SF(5) * PTOT(NK5)
      P1 = SF(3) * PRESS(NK3) + SF(4) * PRESS(NK4) +
     1     SF(5) * PRESS(NK5)
      ENT1 = SF(3) * ENTROP(NK3) + SF(4) * ENTROP(NK4) +
     1     SF(5) * ENTROP(NK5)
      H1 = SF(3) * H(NK3) + SF(4) * H(NK4) +
     1     SF(5) * H(NK5)
      HS1 = SF(3) * HS(NK3) + SF(4) * HS(NK4) +
```

```
     1          SF(5) * HS(NK5)
          K = II
          GOTO 210
 200      RCI1 = RC(NI1)
          VZ1 = VZ(NI1)
          VR1 = VR(NI1)
          VM1 = DSQRT(VZ1 * VZ1 + VR1 * VR1)
          ALP1 = ALP(NI1)
          RHO1 = RHO(NI1)
          RHOT1 = RHOTT(NI1)
          P1 = PRESS(NI1)
          PT1 = PTOT(NI1)
          TT1 = TTOT(NI1)
          T1 = TEMP(NI1)
          H1 = H(NI1)
          HS1 = HS(NI1)
          GOTO 215
 210      IF(NTE1.EQ.1.OR.ISTA1.EQ.3) GOTO 300
 215      IT = 0
          K = II
C               CHECK TO SEE IF P IS WITHIN THE PRESENT ELEMENT
 230      IF(P.GE.PSI(NODE(K,7)))GOTO240
C               CHECK NEXT ELEMENT BELOW THE PRESENT ELEMENT.
          K = K + 1
          GOTO230
 240      IF(P.GT.PSI(NODE(K,8)))GOTO270
          EL = E1(8)
          ER = E1(7)
 250      E2 = (EL + ER)/2.D0
          CALL SHAPE(E2,1.D0,SF)
          PA = SF(1)*PSI(NODE(K,1)) + SF(7)*PSI(NODE(K,7))
     1         + SF(8)*PSI(NODE(K,8))
C               CHECK FOR STREAMLINE CONVERGENCE
          EPS = DABS(PA - P)
          IF(EPS.LT.1.D-06)GOTO290
          IT = IT + 1
          IF(IT.GT.15)GOTO290
          IF(PA.LT.P)GOTO260
          EL = E2
          GOTO250
 260      ER = E2
          GOTO250
 270      IF(P.GT.PSI(NODE(K,1)))GOTO285
 265      EL = E1(1)
          ER = E1(8)
          GOTO250
C               CHECK NEXT ELEMENT ABOVE PRESENT ELEMENT
 285      K = K - 1
          GOTO240
C
C               IF CONVERGENCE CRITERIA SATISFIED.
C               CALCULATE WHIRL AND STATIC ENTHALPY
 290      RCI2 = SF(1)*RC(NODE(K,1)) + SF(7)*RC(NODE(K,7))
     1         + SF(8)*RC(NODE(K,8))
          B2 = SF(1)*B(NODE(K,1)) + SF(7)*B(NODE(K,7))
     1         + SF(8)*B(NODE(K,8))

C               GO TO NEXT ELEMENT.
 300      N1 = NODE(K,1)
          N2 = NODE(K,2)
          N3 = NODE(K,3)
          N4 = NODE(K,4)
          N5 = NODE(K,5)
          N6 = NODE(K,6)
          N7 = NODE(K,7)
          N8 = NODE(K,8)
          RC$(1) = RC(N1)
          RC$(2) = RC(N2)
          RC$(3) = RC(N3)
          RC$(4) = RC(N4)
          RC$(5) = RC(N5)
          RC$(6) = RC(N6)
```

```
      RC$(7) = RC(N7)
      RC$(8) = RC(N8)
      ZC$(1) = ZC(N1)
      ZC$(2) = ZC(N2)
      ZC$(3) = ZC(N3)
      ZC$(4) = ZC(N4)
      ZC$(5) = ZC(N5)
      ZC$(6) = ZC(N6)
      ZC$(7) = ZC(N7)
      ZC$(8) = ZC(N8)
      DO 500 I = 1,NNE
C           FIND THE JACOBIAN.
         CALL JACOB(R1(I),Z1(I),D,E,RC$,ZC$,RJAC)
         DETJ = RJAC(1,1)*RJAC(2,2) - RJAC(1,2)*RJAC(2,1)
C           FIND INVERSE OF JACOBIAN.
         DUM1 = RJAC(1,1) / DETJ
         RJAC(1,1) = RJAC(2,2) / DETJ
         RJAC(1,2) = -RJAC(1,2) / DETJ
         RJAC(2,1) = -RJAC(2,1) / DETJ
         RJAC(2,2) = DUM1
C
C           FIND DNI/DR AND DNI/DZ
C
         DO 400 L = 1,NNE
            DZ(L) = RJAC(1,1)*D(L) + RJAC(1,2)*E(L)
            DR(L) = RJAC(2,1)*D(L) + RJAC(2,2)*E(L)
400      CONTINUE
C
C           CHECK TO SEE IF SOLUTION IS AT INLET
C
C           FIND D(PSI)/DR AND D(PSI)/DZ
         DPSIR(I) = DR(1)*PSI(N1) + DR(2)*PSI(N2) +
     1              DR(3)*PSI(N3) + DR(4)*PSI(N4) +
     2              DR(5)*PSI(N5) + DR(6)*PSI(N6) +
     3              DR(7)*PSI(N7) + DR(8)*PSI(N8)
         DPSIZ(I) = DZ(1)*PSI(N1) + DZ(2)*PSI(N2) +
     1              DZ(3)*PSI(N3) + DZ(4)*PSI(N4) +
     2              DZ(5)*PSI(N5) + DZ(6)*PSI(N6) +
     3              DZ(7)*PSI(N7) + DZ(8)*PSI(N8)
500   CONTINUE
      IF(NTE1.EQ.1) GOTO 600
      IF(ISTA1.EQ.3) GOTO 600
      DPSIR2 = SF(1) * DPSIR(1) + SF(7) * DPSIR(7) +
     1         SF(8) * DPSIR(8)
      DPSIZ2 = SF(1) * DPSIZ(1) + SF(7) * DPSIZ(7) +
     1         SF(8) * DPSIZ(8)
      GOTO 700
600   DPSIR2 = DPSIR(M)
      DPSIZ2 = DPSIZ(M)
700   CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE FCAL(F,W,H,ZA,EA,VZ,RC,ZC,NEL,VU,NBC,NODE,WU,
     1NTE,HR,ENTROP,TTOT,NNOD,NE4,NNE4,NROWS)
C
C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                   ++
C ++       THIS SUBROUTINE CALCULATES THE RIGHT-HAND SIDE VECTOR        ++
C ++       F(R,Z) FROM KNOWN RADIAL DISTRIBUTIONS OF WHIRL,             ++
C ++       ENTHALPY,ROTHALPY, AND ENTROPY.                             ++
C ++        CALL STATEMENT DEFINITIONS:                                ++
C ++             F = RIGHT HAND SIDE VECTOR                            ++
C ++             W = GAUSIAN WEIGHT FUNCTION ARRAY                     ++
C ++             H = NODAL TOTAL ENTHALPY VECTOR                       ++
C ++            ZA = ARRAY OF EXCIE GAUSIAN VALUES                     ++
C ++            EA = ARRAY OF ETA GAUSIAN VALUES                       ++
```

```
C ++            VZ = NODAL AXIAL VELOCITY                                    ++
C ++            RC = ARRAY OF NODAL R COORDINATES                            ++
C ++            ZC = ARRAY OF NODAL Z COORDINATES                            ++
C ++            WRL = NODAL ANGULAR MOMENTUM VECTOR                          ++
C ++            VU = NODAL ABSOLUTE TANGENTIAL VELOCITY VECTOR               ++
C ++            NBC = NODES WHERE BOUNDARY CONDITIONS APPLY                  ++
C ++            NODE = ARRAY OF ELEMENTAL NODE ASSIGNMENTS                   ++
C ++            NN = NUMBER OF NODES IN THE MESH                             ++
C ++            NE = NUMBER OF ELEMENTS IN THE MESH                          ++
C ++            WU = NODAL RELATIVE TANGENTIAL VELOCITY VECTOR               ++
C ++            NTE = TYPE OF ELEMENT ARRAY                                  ++
C ++                                                                         ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNCO,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNPC
      COMMON /NCOUNT/ MROW,MROW1,KK
      COMMON /FCC4/ RG,G,CP,PT,TT,WG,WDOT,RPCT,RHOSTA,
     1LINLET,LOUTLET,PSII,OTJ,F212,FI11,GC
      DIMENSION ZC(1),RC(1),H(1),F(1),TTOT(1)
      DIMENSION VZ(1),VU(1),WRL(1),WU(1),SFI(8)
      DIMENSION NBC(1),NTE(1),NODE(NE4,1)
      DIMENSION W(9),ZA(9),EA(9),F$(8),HR(1),ENTROP(1)
      DIMENSION D(8),E(8),SF(8),RC$(8),ZC$(8),RJAC(2,2)
C
C               ZEROIZE OUT F$() .
C
      DO 50 I = 1,NNE
         F$(I) = 0.DO
50    CONTINUE
C
C               CYCLE FOR EACH ELEMENT.
C
      DO 100 II = 1,NE
         N1 = NODE(II,1)
         N2 = NODE(II,2)
         N3 = NODE(II,3)
         N4 = NODE(II,4)
         N5 = NODE(II,5)
         N6 = NODE(II,6)
         N7 = NODE(II,7)
         N8 = NODE(II,8)
         RC$(1) = RC(N1)
         RC$(2) = RC(N2)
         RC$(3) = RC(N3)
         RC$(4) = RC(N4)
         RC$(5) = RC(N5)
         RC$(6) = RC(N6)
         RC$(7) = RC(N7)
         RC$(8) = RC(N8)
         ZC$(1) = ZC(N1)
         ZC$(2) = ZC(N2)
         ZC$(3) = ZC(N3)
         ZC$(4) = ZC(N4)
         ZC$(5) = ZC(N5)
         ZC$(6) = ZC(N6)
         ZC$(7) = ZC(N7)
         ZC$(8) = ZC(N8)
C
C               CYCLE FOR EACH LOCAL NODE.
C
         DO 300 J = 1,9
            CALL SHAPE(EA(J),ZA(J),SF)
            CALL JACOB(EA(J),ZA(J),D,E,RC$,ZC$,RJAC)
            DETJ = RJAC(1,1)*RJAC(2,2) - RJAC(1,2)*RJAC(2,1)
C             FIND THE INVERSE OF THE JACOBIAN.
         DUM1 = RJAC(1,1) / DETJ
         RJAC(1,1) = RJAC(2,2) / DETJ
```

```
      RJAC(1,2) = -RJAC(1,2) / DETJ
      RJAC(2,1) = -RJAC(2,1) / DETJ
      RJAC(2,2) = DUM1
C
C     FIND NI*LI,NI*VI,THETAI,NI*RI,D(WRL)/DR,D(H)/DR
C
               SUMU = 0.D0
               SUMV = 0.D0
               SUMR = 0.D0
               DSDR = 0.D0
               SUMT = 0.D0
               TDSDR = 0.D0
               DWRLR = 0.D0
               DHDR = 0.D0
               DO 110 KL = 1,NNE
                  N41 = NODE(II,KL)
                  SUMU = SUMU + SF(KL)*VZ(N41)
                  SUMT = SUMT + SF(KL)*TTOT(N41)
                  IF(NTE(II).EQ.2)GOTO105
                  SUMV = SUMV + SF(KL)*VU(N41)
                  DHDR = DHDR + (RJAC(2,1)*D(KL)
     1                 + RJAC(2,2)*E(KL))*H(N41)
                  DSDR = DSDR + (RJAC(2,1)*D(KL)
     1                 + RJAC(2,2)*E(KL))*ENTROP(N41)
                  GOTO106
105               SUMV = SUMV + SF(KL)*VU(N41)
                  DHDR = DHDR + (RJAC(2,1)*D(KL)
     1                 + RJAC(2,2)*E(KL))*HR(N41)
                  DSDR = DSDR + (RJAC(2,1)*D(KL)
     1                 + RJAC(2,2)*E(KL))*ENTROP(N41)
106               SUMR = SUMR + SF(KL)*RC(N41)
                  DWRLR = DWRLR + (RJAC(2,1)*D(KL)
     1                 + RJAC(2,2)*E(KL))*WRL(N41)
110            CONTINUE
C
C                FIND F$(NODE(II,I))
C
         TDSDR = SUMT * DSDR
        DO 200 I = 1,NNE
         XX = (TDSDR - DHDR + (SUMV/SUMR)*DWRLR)*(SF(I)/SUMU)*DETJ
            F$(I) =  F$(I) + XX*W(J)
200         CONTINUE
300      CONTINUE
C
C               ASSEMBLE RIGHT HAND SIDE VECTOR.
C
            F(N1) = F(N1) + F$(1)
            F(N2) = F(N2) + F$(2)
            F(N3) = F(N3) + F$(3)
            F(N4) = F(N4) + F$(4)
            F(N5) = F(N5) + F$(5)
            F(N6) = F(N6) + F$(6)
            F(N7) = F(N7) + F$(7)
            F(N8) = F(N8) + F$(8)
         DO 500 I = 1,NNE
            F$(I) = 0.D0
500      CONTINUE
100   CONTINUE
      RETURN
      END
C
C
C  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  ++                                                                    ++
C  ++     SUBROUTINE CSIMQ (NON-IMSL)                                    ++
C  ++     ADAPTED FOR THE 360 BY RON BRUNELL                             ++
C  ++                                                                    ++
C  ++     PURPOSE                                                        ++
C  ++        OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR             ++
C  ++        EQUATIONS, AX=B                                             ++
C  ++                                                                    ++
```

```
C ++      USAGE                                                              ++
C ++         CALL DSIMQ(A,B,N,KS)                                            ++
C ++                                                                         ++
C ++      DESCRIPTION OF PARAMETERS                                          ++
C ++         A AND B MUST BE REAL*8                                          ++
C ++         A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.                   ++
C ++              THESE ARE DESTROYED IN THE COMPUTATION.  THE               ++
C ++              SIZE OF MATRIX A IS N BY N.++                              ++
C ++         B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE              ++
C ++              ARE REPLACED BY FINAL SOLUTION VALUES, VECTOR X.           ++
C ++         N - NUMBER OF EQUATIONS AND VARIABLES                           ++
C ++         KS - OUTPUT DIGIT                                               ++
C ++               0 FOR A NORMAL SOLUTION                                   ++
C ++               1 FOR A SINGULAR SET OF EQUATIONS                         ++
C ++                                                                         ++
C ++      REMARKS                                                            ++
C ++         MATRIX A MUST BE GENERAL.                                       ++
C ++         IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANING-            ++
C ++         LESS. AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING          ++
C ++         MATRIX INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).             ++
C ++                                                                         ++
C ++      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                      ++
C ++         NONE                                                            ++
C ++                                                                         ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
      SUBROUTINE DSIMQ(A,B,N,KS)
C
C
C     SUBROUTINE DSIMQ NOT INCLUDED NON-IMSL LIBRARY SUBROUTINE
C
C
C
C
      SUBROUTINE STIFF(RC,ZC,B,EM$,ZA.EA,W,NODE,RHO,
     1PSI,F,RHS,EM,NBC,NNOD,NE4,NNE4,NROWS)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                         ++
C ++                                                                         ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /FCOI/ RG,G,CP,PT,TT,WG,WDOT,RHOT,RHOSTA.
     1UINLET,UOUTLET,PSII,BTU.F2I2,F1I1,SC
      COMMON /NCOUNT/ NROW,NROW1,KK
      COMMON /LIO/ NREAD,NWRITE
      DIMENSION  ZC(1),RC(1),B(1),BBB(8),RHO(1)
      DIMENSION NBC(1),NODE(NE4,1),W(8)
      DIMENSION EM(NNOD,1),RIJAC(2,2),PSI(1),F(1),RHS(1)
      DIMENSION D(4),E(8),SF(8),ZA(9),EA(9),W(9),DNDZ(8)
      DIMENSION RC$(8),ZC$(8),EMI(8,8),RJAC(2,2),DNDR(8)
C
C
      DO 400 II = 1,NE
         N1 = NODE(II,1)
         N2 = NODE(II,2)
         N3 = NODE(II,3)
         N4 = NODE(II,4)
         N5 = NODE(II,5)
         N6 = NODE(II,6)
         N7 = NODE(II,7)
         N8 = NODE(II,8)
         RC$(1) = RC(N1)
         RC$(2) = RC(N2)
         RC$(3) = RC(N3)
         RC$(4) = RC(N4)
```

```
          RC$(5) = RC(N5)
          RC$(6) = RC(N6)
          RC$(7) = RC(N7)
          RC$(8) = RC(N8)
          ZC$(1) = ZC(N1)
          ZC$(2) = ZC(N2)
          ZC$(3) = ZC(N3)
          ZC$(4) = ZC(N4)
          ZC$(5) = ZC(N5)
          ZC$(6) = ZC(N6)
          ZC$(7) = ZC(N7)
          ZC$(8) = ZC(N8)
C
C
C              PERFORM GAUSSIAN QUADRATURE INTEGRATION
C
C
          DO 320 I = 1,9
             CALL SHAPE(EA(I),ZA(I),SF)
             CALL JACOB(EA(I),ZA(I),D,E,RC$,ZC$,RJAC)
             DETJ = RJAC(1,1)*RJAC(2,2) - RJAC(1,2)*RJAC(2,1)
C                 FIND INVERSE OF JACOBIAN
             DUM1 = RJAC(1,1) / DETJ
             RJAC(1,1) = RJAC(2,2) / DETJ
             RJAC(1,2) = -RJAC(1,2) / DETJ
             RJAC(2,1) = -RJAC(2,1) / DETJ
             RJAC(2,2) = DUM1
             DO 321 J = 1,8
                DNDZ(J) = RJAC(1,1)*D(J) + RJAC(1,2)*E(J)
                DNDR(J) = RJAC(2,1)*D(J) + RJAC(2,2)*E(J)
321          CONTINUE
          IF(II.NE.1) GOTO 100
100       CONTINUE
C
C                 FIND RHO, R, AND B FOR NUMERICAL INTEGRATION.
C
             RHORB = 0.0D0
             DO 330 L = 1,NNE
                NIIL = NODE(II,L)
                RHORB = RHORB + SF(L)*RHO(NIIL)*RC(NIIL)*B(NIIL)
330          CONTINUE
             SMK = (1.D0/(RHORB))*144.D0)*12.D00
C
          DO 300 J = 1,NNE
          IF(II.NE.1) GOTO 309
309       CONTINUE
             DO 310 K = 1,NNE
                   EM$(J,K) = EM$(J,K) + W(I) * SMK * (DNDZ(J)*DNDZ(K)
     1                        + DNDR(J)*DNDR(K)) * DETJ
C    2                        * 144.D0
310          CONTINUE
300       CONTINUE
320    CONTINUE
C
C          ASSEMBLE SYSTEM INFLUENCE MATRIX W/OUT REGARD FOR
C          BOUNDARY CONDITIONS
          N(1) = N1
          N(2) = N2
          N(3) = N3
          N(4) = N4
          N(5) = N5
          N(6) = N6
          N(7) = N7
          N(8) = N8
          DO 350 I$ = 1,NNE
             I = N(I$)
             DO 350 J$ = 1,NNE
             J = N(J$)
             EM(I,J) = EM(I,J) + EM$(I$,J$)
350       CONTINUE
C
C                 ZEROIZE OUT EM$( ) FOR NEXT ELEMENT
```

```
          DO 370 I2 = 1,NNE
             DO 370 J2 = 1,NNE
             EM1(I2,J2) = 0.D0
370       CONTINUE
C
C                RECYCLE FOR NEXT ELEMENT
400   CONTINUE
C
C          MODIFY SYSTEM OF EQUATIONS TO INCLUDE BOUNDARY CONDITIONS
C
      DO 410 I = 1,NN
         DO 410 J = 1,NNBC
            JX = NBC(J)
            F(I) = F(I) - EM(I,JX)*PSI(JX)
            EM(I,JX)= 0.D0
            EM(JX,I)= 0.D0
            EM(JX,JX)= 1.D0
            F(JX) = PSI(JX)
410   CONTINUE
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE REPLA(PSI,F,RHS,NNOD,NE4,NNE4)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                     ++
C ++                                                                     ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /NCOUNT/ NROW,NROW1,KK
      DIMENSION PSI(1),PSIO(1),F(1),RHS(1)
C
C   REPLACE PSI(I) WITH SOLUTION VECTOR AND RESET F(I) WITH RHS(I)
C
      DO 100 I = 1,NN
         PSI(I) = F(I)
         F(I) = RHS(I)
100   CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE RELAX(RELX,PSI,PSIO,NBC,NNOD,NE4,NNE4)
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                     ++
C ++                                                                     ++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /NCOUNT/ NROW,NROW1,KK
      DIMENSION PSI(1),PSIO(1),F(1),RHS(1)
      DIMENSION NBC(1)
C
C             IF KK GE 1, PERFORM UNDER RELAXATION BEFORE
C          COMPUTING NEW VELOCITY AND DENSITY DISTRIBUTION.
```

```
C
      DO 100 I = 1,NN
         DO 200 J = 1,NNBC
               LTEST = I - NBC(J)
               IF(LTEST.EQ.0) GOTO 100
200      CONTINUE
         PSI(I) = PSIO(I) + RELX*(PSI(I) - PSIO(I))
100   CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE NOCON(X,F,PSI,PSIO,EM,IFL,NNOD,NE4,NNE4)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      COMMON /NCOUNT/ NROW,NROW1,KK
      COMMON /LIO/ NREAD,NWRITE
      DIMENSION PSI(1),PSIO(1),EM(NNOD,1),F(1)
      WRITE (NWRITE,1400)KK,X
1400  FORMAT(' ','LARGEST EPS FOR ITERATION ',I2,' IS ',D19.12)
      IF (KK.LT.22)GOTO1500
      IFL = 0
      GOTO 451
1500  WRITE(NWRITE,1102)KK
1102  FORMAT(' ','ITERATION NO. ',I3,' COMPLETE',/,' STREAM FUNCTION CON
     1VERGENCE NOT YET SATISFIED.',/,' NEXT ITERATION IS IN PROGRESS')
C
C                 PREPARE FOR NEXT ITERATION. ZEROIZE
C                 STIFFNESS MATRIX AND RIGHT HAND SIDE VECTOR
C                 REPLACE PSIO() WITH CURRENT VALUE OF PSI.
C
      IFL = 2
      DO 460 I = 1,NN
         F(I) = 0.D0
         PSIO(I) = PSI(I)
         DO 460 J = 1,NN
            EM(I,J) = 0.D0
460   CONTINUE
451   RETURN
      END
C
C
C
C
C
      SUBROUTINE TEST(PSI,PSIO,X,NNOD,NE4,NNE4,NROWS)
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++                                                                  ++
C ++                                                                  ++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      IMPLICIT REAL*8(A-H,P-Z)
      INTEGER*4 NREAD,NWRITE,IC,NNOD,NE4,NNE4,NROWS,LIMR,LIMI
      COMMON /NCOUNT/ NCOL,NCOL1,NN,NE
     1,NNE,NNBC,NNNBC
      DIMENSION PSI(1),PSIO(1)
C
C   COMPARE NEW AND OLD STREAM FUNCTION DISTRIBUTIONS
C
      X = 0.D0
```

```
      DO 100 I = 1,NN
         IF(PSI(I).EQ.0.D0) GOTO 110
         EPS = DABS((PSIO(I) - PSI(I))/PSI(I))
         GOTO120
110      EPS = DABS(PSIO(I) - PSI(I))
120      IF (X.GT.EPS) GOTO 100
         X = EPS
100   CONTINUE
      RETURN
      END
```

APPENDIX G

SAMPLE PROGRAM OUTPUT

```
MEMORY SPACE AVAILABLE :
REAL 8 =  409    INTEGER =   988    REAL 4 =  424
```

NASA TASK-1 TRANSONIC COMPRESSOR

NO. OF NODES = 222 NO. OF ELEMENTS = 63

NO. OF ROWS = 7 NO. OF COLUMNS = 9

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
1	0.0	0.188780D+02	0.910000D+00	0.0	0.0
2	0.0	0.182900D+02	0.910000D+00	0.0	0.0
3	0.0	0.176824D+02	0.910000D+00	0.0	0.0
4	0.0	0.170532D+02	0.910000D+00	0.0	0.0
5	0.0	0.163999D+02	0.910000D+00	0.0	0.0
6	0.0	0.157194D+02	0.910000D+00	0.0	0.0
7	0.0	0.150081D+02	0.910000D+00	0.0	0.0
8	0.0	0.142614D+02	0.910000D+00	0.0	0.0
9	0.0	0.134734D+02	0.910000D+00	0.0	0.0
10	0.0	0.126343D+02	0.910000D+00	0.0	0.0
11	0.0	0.117397D+02	0.910000D+00	0.0	0.0
12	0.0	0.107696D+02	0.910000D+00	0.0	0.0
13	0.0	0.970091D+01	0.910000D+00	0.0	0.0
14	0.0	0.850010D+01	0.910000D+00	0.0	0.0
15	0.0	0.709900D+01	0.910000D+00	0.0	0.0
16	0.150000D+01	0.186510D+02	0.910000D+00	0.0	0.0
17	0.150000D+01	0.175021D+02	0.910000D+00	0.0	0.0
18	0.150000D+01	0.162379D+02	0.910000D+00	0.0	0.0
19	0.150000D+01	0.148660D+02	0.910000D+00	0.0	0.0
20	0.150000D+01	0.133552D+02	0.910000D+00	0.0	0.0
21	0.150000D+01	0.116493D+02	0.910000D+00	0.0	0.0
22	0.150000D+01	0.964623D+01	0.910000D+00	0.0	0.0
23	0.150000D+01	0.709900D+01	0.910000D+00	0.0	0.0
24	0.300000D+01	0.184340D+02	0.910000D+00	0.0	0.0
25	0.300000D+01	0.179124D+02	0.910000D+00	0.0	0.0
26	0.300000D+01	0.173219D+02	0.910000D+00	0.0	0.0
27	0.300000D+01	0.167106D+02	0.910000D+00	0.0	0.0
28	0.300000D+01	0.160761D+02	0.910000D+00	0.0	0.0
29	0.300000D+01	0.154154D+02	0.910000D+00	0.0	0.0
30	0.300000D+01	0.147252D+02	0.910000D+00	0.0	0.0
31	0.300000D+01	0.140010D+02	0.910000D+00	0.0	0.0
32	0.300000D+01	0.132372D+02	0.910000D+00	0.0	0.0
33	0.300000D+01	0.124265D+02	0.910000D+00	0.0	0.0
34	0.300000D+01	0.115591D+02	0.910000D+00	0.0	0.0
35	0.300000D+01	0.106212D+02	0.910000D+00	0.0	0.0
36	0.300000D+01	0.959191D+01	0.910000D+00	0.0	0.0
37	0.300000D+01	0.843802D+01	0.910000D+00	0.0	0.0
38	0.300000D+01	0.709900D+01	0.910000D+00	0.0	0.0
39	0.450000D+01	0.184460D+02	0.910000D+00	0.0	0.0
40	0.450000D+01	0.172972D+02	0.910000D+00	0.0	0.0

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
41	0.450000D+01	0.160449D+02	0.910000D+00	0.0	0.0
42	0.450000D+01	0.146979D+02	0.910000D+00	0.0	0.0
43	0.450000D+01	0.132144D+02	0.910000D+00	0.0	0.0
44	0.450000D+01	0.115418D+02	0.910000D+00	0.0	0.0
45	0.450000D+01	0.958145D+01	0.910000D+00	0.0	0.0
46	0.450000D+01	0.709900D+01	0.910000D+00	0.0	0.0
47	0.600000D+01	0.184080D+02	0.910000D+00	0.0	0.0
48	0.600000D+01	0.178336D+02	0.910000D+00	0.0	0.0
49	0.600000D+01	0.172524D+02	0.910000D+00	0.0	0.0
50	0.600000D+01	0.166446D+02	0.910000D+00	0.0	0.0
51	0.600000D+01	0.160137D+02	0.910000D+00	0.0	0.0
52	0.600000D+01	0.153569D+02	0.910000D+00	0.0	0.0
53	0.600000D+01	0.146707D+02	0.910000D+00	0.0	0.0
54	0.600000D+01	0.139508D+02	0.910000D+00	0.0	0.0
55	0.600000D+01	0.131917D+02	0.910000D+00	0.0	0.0
56	0.600000D+01	0.123362D+02	0.910000D+00	0.0	0.0
57	0.600000D+01	0.115244D+02	0.910000D+00	0.0	0.0
58	0.600000D+01	0.105928D+02	0.910000D+00	0.0	0.0
59	0.600000D+01	0.957101D+01	0.910000D+00	0.0	0.0
60	0.600000D+01	0.842615D+01	0.910000D+00	0.0	0.0
61	0.600000D+01	0.709900D+01	0.910000D+00	0.0	0.0
62	0.890200D+01	0.184025D+02	0.910000D+00	0.0	0.0
63	0.890200D+01	0.177474D+02	0.910000D+00	0.0	0.0
64	0.890200D+01	0.160072D+02	0.910000D+00	0.0	0.0
65	0.890200D+01	0.146669D+02	0.910000D+00	0.0	0.0
66	0.890200D+01	0.131886D+02	0.910000D+00	0.0	0.0
67	0.890200D+01	0.115221D+02	0.910000D+00	0.0	0.0
68	0.890200D+01	0.955981D+01	0.910000D+00	0.0	0.0
69	0.890200D+01	0.709950D+01	0.910000D+00	0.0	0.0
70	0.118040D+02	0.183970D+02	0.910000D+00	0.0	0.0
71	0.118040D+02	0.178291D+02	0.910000D+00	0.0	0.0
72	0.118040D+02	0.172424D+02	0.910000D+00	0.0	0.0
73	0.118040D+02	0.166351D+02	0.910000D+00	0.0	0.0
74	0.118040D+02	0.160046D+02	0.910000D+00	0.0	0.0
75	0.118040D+02	0.153486D+02	0.910000D+00	0.0	0.0
76	0.118040D+02	0.146630D+02	0.910000D+00	0.0	0.0
77	0.118040D+02	0.139438D+02	0.910000D+00	0.0	0.0
78	0.118040D+02	0.131854D+02	0.910000D+00	0.0	0.0
79	0.118040D+02	0.123807D+02	0.910000D+00	0.0	0.0
80	0.118040D+02	0.115199D+02	0.910000D+00	0.0	0.0

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
81	0.118040D+02	0.105993D+02	0.910000D+00	0.0	0.0
82	0.118040D+02	0.955862D+01	0.910000D+00	0.0	0.0
83	0.118040D+02	0.942522D+01	0.910000D+00	0.0	0.0
84	0.118040D+02	0.710000D+01	0.910000D+00	0.0	0.0
85	0.128915D+02	0.183935D+02	0.910000D+00	0.0	0.0
86	0.128915D+02	0.172556D+02	0.910000D+00	0.0	0.0
87	0.128915D+02	0.160394D+02	0.910000D+00	0.0	0.0
88	0.128915D+02	0.147218D+02	0.910000D+00	0.0	0.0
89	0.128915D+02	0.132741D+02	0.910000D+00	0.0	0.0
90	0.128915D+02	0.116473D+02	0.910000D+00	0.0	0.0
91	0.128915D+02	0.975407D+01	0.910000D+00	0.0	0.0
92	0.128915D+02	0.739000D+01	0.910000D+00	0.0	0.0
93	0.139790D+02	0.183900D+02	0.910000D+00	0.0	0.0
94	0.139790D+02	0.178395D+02	0.910000D+00	0.0	0.0
95	0.139790D+02	0.172715D+02	0.910000D+00	0.0	0.0
96	0.139790D+02	0.166842D+02	0.910000D+00	0.0	0.0
97	0.139790D+02	0.160754D+02	0.910000D+00	0.0	0.0
98	0.139790D+02	0.154426D+02	0.910000D+00	0.0	0.0
99	0.139790D+02	0.147928D+02	0.910000D+00	0.0	0.0
100	0.139790D+02	0.140921D+02	0.910000D+00	0.0	0.0
101	0.139790D+02	0.133655D+02	0.910000D+00	0.0	0.0
102	0.139790D+02	0.125976D+02	0.910000D+00	0.0	0.0
103	0.139790D+02	0.117795D+02	0.910000D+00	0.0	0.0
104	0.139790D+02	0.109001D+02	0.910000D+00	0.0	0.0
105	0.139790D+02	0.994331D+01	0.910000D+00	0.0	0.0
106	0.139790D+02	0.838403D+01	0.910000D+00	0.0	0.0
107	0.139790D+02	0.758000D+01	0.910000D+00	0.0	0.0
108	0.162450D+02	0.182900D+02	0.910000D+00	0.0	0.0
109	0.162085D+02	0.172285D+02	0.910000D+00	0.0	0.0
110	0.161720D+02	0.160971D+02	0.910000D+00	0.0	0.0
111	0.161355D+02	0.148000D+02	0.910000D+00	0.0	0.0
112	0.160990D+02	0.135540D+02	0.910000D+00	0.0	0.0
113	0.160625D+02	0.120334D+02	0.910000D+00	0.0	0.0
114	0.160260D+02	0.104070D+02	0.910000D+00	0.0	0.0
115	0.159895D+02	0.840350D+01	0.910000D+00	0.0	0.0
116	0.185110D+02	0.181900D+02	0.902868D+00	0.0	0.0
117	0.184745D+02	0.176972D+02	0.902443D+00	0.0	0.0
118	0.184380D+02	0.171302D+02	0.901964D+00	0.0	0.0
119	0.184015D+02	0.166678D+02	0.901440D+00	0.0	0.0
120	0.183650D+02	0.161286D+02	0.900852D+00	0.0	0.0

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
121	0.183295D+02	0.155706D+02	0.900197D+00	0.0	0.0
122	0.182920D+02	0.149159D+02	0.899463D+00	0.0	0.0
123	0.182555D+02	0.143900D+02	0.898637D+00	0.0	0.0
124	0.182190D+02	0.137617D+02	0.897699D+00	0.0	0.0
125	0.181825D+02	0.131034D+02	0.896626D+00	0.0	0.0
126	0.181460D+02	0.124101D+02	0.895388D+00	0.0	0.0
127	0.181095D+02	0.116758D+02	0.893946D+00	0.0	0.0
128	0.180730D+02	0.108921D+02	0.892245D+00	0.0	0.0
129	0.180365D+02	0.100475D+02	0.890219D+00	0.0	0.0
130	0.180000D+02	0.912500D+01	0.887743D+00	0.0	0.0
131	0.194405D+02	0.180305D+02	0.870122D+00	0.0	0.0
132	0.194537D+02	0.170855D+02	0.860320D+00	0.0	0.0
133	0.194669D+02	0.160850D+02	0.849791D+00	0.0	0.0
134	0.194801D+02	0.150190D+02	0.835509D+00	0.0	0.0
135	0.194934D+02	0.138692D+02	0.819544D+00	0.0	0.0
136	0.195066D+02	0.126162D+02	0.798639D+00	0.0	0.0
137	0.195198D+02	0.112241D+02	0.762254D+00	0.0	0.0
138	0.195330D+02	0.963500D+01	0.722375D+00	0.0	0.0
139	0.203700D+02	0.178710D+02	0.903041D+00	0.0	0.0
140	0.204197D+02	0.174329D+02	0.902503D+00	0.0	0.0
141	0.204694D+02	0.169835D+02	0.901992D+00	0.0	0.0
142	0.205191D+02	0.165219D+02	0.901445D+00	0.0	0.0
143	0.205689D+02	0.160470D+02	0.900855D+00	0.0	0.0
144	0.206186D+02	0.155576D+02	0.900204D+00	0.0	0.0
145	0.206683D+02	0.150523D+02	0.899478D+00	0.0	0.0
146	0.207180D+02	0.145295D+02	0.898668D+00	0.0	0.0
147	0.207677D+02	0.139671D+02	0.897793D+00	0.0	0.0
148	0.208174D+02	0.134229D+02	0.896796D+00	0.0	0.0
149	0.208671D+02	0.128338D+02	0.895680D+00	0.0	0.0
150	0.209169D+02	0.122164D+02	0.894372D+00	0.0	0.0
151	0.209666D+02	0.115660D+02	0.892748D+00	0.0	0.0
152	0.210163D+02	0.108769D+02	0.890641D+00	0.0	0.0
153	0.210660D+02	0.101410D+02	0.887700D+00	0.0	0.0
154	0.216100D+02	0.178535D+02	0.910000D+00	0.0	0.0
155	0.216807D+02	0.169845D+02	0.910000D+00	0.0	0.0
156	0.217514D+02	0.160791D+02	0.910000D+00	0.0	0.0
157	0.218224D+02	0.151140D+02	0.910000D+00	0.0	0.0
158	0.218931D+02	0.140829D+02	0.910000D+00	0.0	0.0
159	0.219639D+02	0.129701D+02	0.910000D+00	0.0	0.0
160	0.220347D+02	0.117524D+02	0.910000D+00	0.0	0.0

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
161	0.221055D+02	0.103930D+02	0.910000D+00	0.0	0.0
162	0.229300D+02	0.178360D+02	0.899068D+00	0.0	0.0
163	0.228711D+02	0.174211D+02	0.898932D+00	0.0	0.0
164	0.228921D+02	0.169950D+02	0.898775D+00	0.0	0.0
165	0.229132D+02	0.165601D+02	0.898593D+00	0.0	0.0
166	0.229343D+02	0.161122D+02	0.898383D+00	0.0	0.0
167	0.229554D+02	0.156518D+02	0.898143D+00	0.0	0.0
168	0.229764D+02	0.151772D+02	0.897873D+00	0.0	0.0
169	0.229975D+02	0.146874D+02	0.897572D+00	0.0	0.0
170	0.230186D+02	0.141806D+02	0.897240D+00	0.0	0.0
171	0.230396D+02	0.136551D+02	0.896878D+00	0.0	0.0
172	0.230607D+02	0.131085D+02	0.896483D+00	0.0	0.0
173	0.230819D+02	0.125381D+02	0.896051D+00	0.0	0.0
174	0.231029D+02	0.119404D+02	0.895573D+00	0.0	0.0
175	0.231239D+02	0.113113D+02	0.895043D+00	0.0	0.0
176	0.231450D+02	0.106450D+02	0.894437D+00	0.0	0.0
177	0.246250D+02	0.178360D+02	0.821289D+00	0.0	0.0
178	0.246161D+02	0.170166D+02	0.821773D+00	0.0	0.0
179	0.246071D+02	0.161557D+02	0.822407D+00	0.0	0.0
180	0.245982D+02	0.152462D+02	0.822935D+00	0.0	0.0
181	0.245893D+02	0.142783D+02	0.823342D+00	0.0	0.0
182	0.245804D+02	0.132412D+02	0.823361D+00	0.0	0.0
183	0.245714D+02	0.121149D+02	0.822973D+00	0.0	0.0
184	0.245625D+02	0.108725D+02	0.821859D+00	0.0	0.0
185	0.264000D+02	0.178360D+02	0.899110D+00	0.0	0.0
186	0.263700D+02	0.174413D+02	0.898892D+00	0.0	0.0
187	0.263400D+02	0.170376D+02	0.898731D+00	0.0	0.0
188	0.263100D+02	0.166240D+02	0.898562D+00	0.0	0.0
189	0.262800D+02	0.161948D+02	0.898390D+00	0.0	0.0
190	0.262500D+02	0.157642D+02	0.898195D+00	0.0	0.0
191	0.262200D+02	0.153163D+02	0.897962D+00	0.0	0.0
192	0.261900D+02	0.148548D+02	0.897579D+00	0.0	0.0
193	0.261600D+02	0.143786D+02	0.897339D+00	0.0	0.0
194	0.261300D+02	0.138860D+02	0.896973D+00	0.0	0.0
195	0.261000D+02	0.133753D+02	0.896535D+00	0.0	0.0
196	0.260700D+02	0.128443D+02	0.896061D+00	0.0	0.0
197	0.260400D+02	0.122904D+02	0.895550D+00	0.0	0.0
198	0.260100D+02	0.117104D+02	0.894991D+00	0.0	0.0
199	0.259800D+02	0.111000D+02	0.894332D+00	0.0	0.0
200	0.259700D+02	0.178360D+02	0.910000D+00	0.0	0.0

SUMMARY OF NODAL COORDINATES

NODE	Z(I)	R(I)	B(I)	ABS FLOW ANG	REL FLOW ANG
201	0.2967000D+02	0.1704080D+02	0.9100000D+00	0.0	0.0
202	0.2964000D+02	0.1620670D+02	0.9100000D+00	0.0	0.0
203	0.2961000D+02	0.1533720D+02	0.9100000D+00	0.0	0.0
204	0.2958000D+02	0.1439410D+02	0.9100000D+00	0.0	0.0
205	0.2955000D+02	0.1339610D+02	0.9100000D+00	0.0	0.0
206	0.2952000D+02	0.1231750D+02	0.9100000D+00	0.0	0.0
207	0.2949000D+02	0.1113500D+02	0.9100000D+00	0.0	0.0
208	0.3300000D+02	0.1783600D+02	0.9100000D+00	0.0	0.0
209	0.3300000D+02	0.1744450D+02	0.9100000D+00	0.0	0.0
210	0.3300000D+02	0.1704410D+02	0.9100000D+00	0.0	0.0
211	0.3300000D+02	0.1663400D+02	0.9100000D+00	0.0	0.0
212	0.3300000D+02	0.1621350D+02	0.9100000D+00	0.0	0.0
213	0.3300000D+02	0.1579190D+02	0.9100000D+00	0.0	0.0
214	0.3300000D+02	0.1533810D+02	0.9100000D+00	0.0	0.0
215	0.3300000D+02	0.1488110D+02	0.9100000D+00	0.0	0.0
216	0.3300000D+02	0.1440950D+02	0.9100000D+00	0.0	0.0
217	0.3300000D+02	0.1392210D+02	0.9100000D+00	0.0	0.0
218	0.3300000D+02	0.1341590D+02	0.9100000D+00	0.0	0.0
219	0.3300000D+02	0.1289190D+02	0.9100000D+00	0.0	0.0
220	0.3300000D+02	0.1234470D+02	0.9100000D+00	0.0	0.0
221	0.3300000D+02	0.1177200D+02	0.9100000D+00	0.0	0.0
222	0.3300000D+02	0.1117000D+02	0.9100000D+00	0.0	0.0

SYSTEM TOPOLOGY

ELEMENT	NODES								TYPE OF ELEMENT
1	24	16	1	2	3	17	26	25	1
2	26	17	3	4	5	18	28	27	1
3	28	18	5	6	7	19	30	29	1
4	30	19	7	8	9	20	32	31	1
5	32	20	9	10	11	21	34	33	1
6	34	21	11	12	13	22	36	35	1
7	36	22	13	14	15	23	38	37	1
8	47	39	24	25	26	40	49	48	1
9	49	40	26	27	28	41	51	50	1
10	51	41	28	29	30	42	53	52	1
11	53	42	30	31	32	43	55	54	1
12	55	43	32	33	34	44	57	56	1
13	57	44	34	35	36	45	59	58	1
14	59	45	36	37	38	46	61	60	1
15	70	62	47	48	49	63	72	71	1
16	72	63	49	50	51	64	74	73	1
17	74	64	51	52	53	65	76	75	1
18	76	65	53	54	55	66	78	77	1
19	78	66	55	56	57	67	80	79	1
20	80	67	57	58	59	68	82	81	1
21	82	68	59	60	61	69	84	83	1
22	93	85	70	71	72	86	95	94	1
23	95	86	72	73	74	87	97	96	1
24	97	87	74	75	76	88	99	98	1
25	99	88	76	77	78	89	101	100	1
26	101	89	78	79	80	90	103	102	1
27	103	90	80	81	82	91	105	104	1
28	105	91	82	83	84	92	107	106	1
29	116	108	93	94	95	109	118	117	1
30	118	109	95	96	97	110	120	119	1
31	120	110	97	98	99	111	122	121	1
32	122	111	99	100	101	112	124	123	1
33	124	112	101	102	103	113	126	125	1
34	126	113	103	104	105	114	128	127	1
35	128	114	105	106	107	115	130	129	1
36	139	131	116	117	118	132	141	140	2
37	141	132	118	119	120	133	143	142	2
38	143	133	120	121	122	134	145	144	2
39	145	134	122	123	124	135	147	146	2
40	147	135	124	125	126	136	149	148	2
41	149	136	126	127	128	137	151	150	2
42	151	137	128	129	130	138	153	152	2
43	162	154	139	140	141	155	164	163	1
44	164	155	141	142	143	156	166	165	1
45	166	156	143	144	145	157	168	167	1
46	168	157	145	146	147	158	170	169	1
47	170	158	147	148	149	159	172	171	1
48	172	159	149	150	151	160	174	173	1
49	174	160	151	152	153	161	176	175	1
50	185	177	162	163	164	178	187	186	3
51	187	178	164	165	166	179	189	188	3
52	189	179	166	167	168	180	191	190	3
53	191	180	168	169	170	181	193	192	3
54	193	181	170	171	172	182	195	194	3
55	195	182	172	173	174	183	197	196	3
56	197	183	174	175	176	184	199	198	3
57	208	200	185	186	187	201	210	209	1
58	210	201	187	188	189	202	212	211	1
59	212	202	189	190	191	203	214	213	1
60	214	203	191	192	193	204	216	215	1
61	216	204	193	194	195	205	218	217	1
62	218	205	195	196	197	206	220	219	1
63	220	206	197	198	199	207	222	221	1

INLET THERMODYNAMIC VARIABLES ARE AS FOLLOWS

```
FLOW RATE =  0.174540D+03 LBM/SEC
TOT DENSITY =  0.851884D-01LBM CU FT
TOT PRESSURE =  0.175000D+02 PSI
TOT TEMPERATURE =  0.555000D+03 DEG RANKINE
ROTATIONAL SPEED =  0.697520D+04 RPM
INLET U VELOCITY =  0.306922D+03 FT/SEC
GAS CONSTANT =  0.533000D+02
RATIO OF SPECIFIC HEATS =  0.140000D+01
SPECIFIC HEAT AT CONSTANT PRESSURE =  0.240000D+00
STATIC DENSITY AT INLET =  0.322124D-01
```

NODES WHERE PSI IS SPECIFIED

NODE NO.	PSI(I)
1	0.277789D+02
2	0.257947D+02
3	0.238105D+02
4	0.218263D+02
5	0.198421D+02
6	0.178579D+02
7	0.158737D+02
8	0.138895D+02
9	0.119052D+02
10	0.992104D+01
11	0.793683D+01
12	0.595262D+01
13	0.396841D+01
14	0.198421D+01
15	0.0
16	0.277789D+02
23	0.0
24	0.277789D+02
38	0.0
39	0.277789D+02
46	0.0
47	0.277789D+02
61	0.0
62	0.277789D+02
69	0.0
70	0.277789D+02
84	0.0
85	0.277789D+02
92	0.0
93	0.277789D+02
107	0.0
108	0.277789D+02
115	0.0
116	0.277789D+02
130	0.0
131	0.277789D+02
138	0.0
139	0.277789D+02
153	0.0
154	0.277789D+02
161	0.0
162	0.277789D+02
176	0.0
177	0.277789D+02
184	0.0
185	0.277789D+02
199	0.0
200	0.277789D+02
207	0.0
208	0.277789D+02
222	0.0

```
LARGEST EPS FOR ITERATION  1 IS  0.3467595582060+00
ITERATION NO.    1 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  2 IS  0.2048022736240+00
ITERATION NO.    2 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  3 IS  0.1359221445230+00
ITERATION NO.    3 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  4 IS  0.9414268241000-01
ITERATION NO.    4 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  5 IS  0.6621886404890-01
ITERATION NO.    5 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  6 IS  0.4694987938000-01
ITERATION NO.    6 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  7 IS  0.3423782590920-01
ITERATION NO.    7 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
LARGEST EPS FOR ITERATION  8 IS  0.2543172791210-01
ITERATION NO.    8 COMPLETE
STREAM FUNCTION CONVERGENCE NOT YET SATISFIED.
NEXT ITERATION IS IN PROGRESS
PROGRAM TERMINATED ON ITERATION NO.  9
RESULTS WHICH FOLLOW ARE FOR CONVERGENCE EPSILON =  0.1890177416520-01
STREAM FUNCTION CONVERGENCE CRITERION SATISFIED ON ITERATION  NUMBER   9

RESULTS ARE AS FOLLOWS FOR CONVERGENCE   EPSILON =  0.1890177416520-01
```

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
1	0.277789D+02	0.306922D+03	0.0	0.188760D+02	0.822124D-01
2	0.257947D+02	0.306922D+03	0.0	0.182900D+02	0.822124D-01
3	0.238105D+02	0.306922D+03	0.0	0.176824D+02	0.822124D-01
4	0.218263D+02	0.306922D+03	0.0	0.170532D+02	0.822124D-01
5	0.198421D+02	0.306922D+03	0.0	0.163959D+02	0.822124D-01
6	0.178579D+02	0.306922D+03	0.0	0.157194D+02	0.822124D-01
7	0.158737D+02	0.306922D+03	0.0	0.150031D+02	0.822124D-01
8	0.138895D+02	0.306922D+03	0.0	0.142614D+02	0.822124D-01
9	0.119052D+02	0.306922D+03	0.0	0.134734D+02	0.822124D-01
10	0.992104D+01	0.306922D+03	0.0	0.126363D+02	0.822124D-01
11	0.793683D+01	0.306922D+03	0.0	0.117397D+02	0.822124D-01
12	0.595262D+01	0.306922D+03	0.0	0.107686D+02	0.822124D-01
13	0.396841D+01	0.306922D+03	0.0	0.970091D+01	0.822124D-01
14	0.198421D+01	0.306922D+03	0.0	0.850010D+01	0.822124D-01
15	0.0	0.306922D+03	0.0	0.709900D+01	0.822124D-01
16	0.277789D+02	0.373509D+03	-0.490541D+02	0.186810D+02	0.807240D-01
17	0.237500D+02	0.371572D+03	-0.360279D+02	0.175021D+02	0.808030D-01
18	0.197270D+02	0.367484D+03	-0.277484D+02	0.162379D+02	0.809130D-01
19	0.157401D+02	0.364755D+03	-0.211303D+02	0.148666D+02	0.909850D-01
20	0.117801D+02	0.363261D+03	-0.155310D+02	0.133552D+02	0.810251D-01
21	0.783993D+01	0.363386D+03	-0.105988D+02	0.116493D+02	0.810263D-01
22	0.391533D+01	0.367943D+03	-0.605567D+01	0.964629D+01	0.809253D-01
23	0.0	0.367090D+03	0.0	0.709900D+01	0.809458D-01
24	0.277789D+02	0.406134D+03	-0.533389D+02	0.183440D+02	0.799256D-01
25	0.256598D+02	0.393288D+03	-0.411491D+02	0.179124D+02	0.801554D-01
26	0.235723D+02	0.385405D+03	-0.290222D+02	0.173219D+02	0.804932D-01
27	0.215454D+02	0.380336D+03	-0.254536D+02	0.167106D+02	0.806192D-01
28	0.195374D+02	0.377951D+03	-0.228207D+02	0.160761D+02	0.806791D-01
29	0.175432D+02	0.373890D+03	-0.200106D+02	0.154154D+02	0.807773D-01
30	0.155577D+02	0.373221D+03	-0.178725D+02	0.147252D+02	0.807952D-01
31	0.135513D+02	0.369333D+03	-0.155950D+02	0.140030D+02	0.808361D-01
32	0.116295D+02	0.370150D+03	-0.134974D+02	0.132372D+02	0.809703D-01
33	0.967579D+01	0.355850D+03	-0.110530D+02	0.124265D+02	0.809740D-01
34	0.772930D+01	0.369019D+03	-0.932399D+01	0.115591D+02	0.809991D-01
35	0.578925D+01	0.362052D+03	-0.690637D+01	0.106212D+02	0.810583D-01
36	0.385730D+01	0.372631D+03	-0.545479D+01	0.959191D+01	0.805179D-01
37	0.192490D+01	0.357826D+03	-0.235355D+01	0.843802D+01	0.311540D-01
38	0.0	0.370191D+03	-0.547992D-13	0.709900D+01	0.808750D-01
39	0.277789D+02	0.393266D+03	-0.995273D+01	0.184460D+02	0.805272D-01
40	0.236837D+02	0.388745D+03	-0.128843D+02	0.172872D+02	0.804342D-01

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
41	0.196246D+02	0.382613D+03	-0.116983D+02	0.160445D+02	0.805813D-01
42	0.156204D+02	0.377925D+03	-0.945150D+01	0.146979D+02	0.806931D-01
43	0.116615D+02	0.374427D+03	-0.680554D+01	0.132144D+02	0.807759D-01
44	0.774100D+01	0.372201D+03	-0.413513D+01	0.115418D+02	0.808282D-01
45	0.385562D+01	0.371532D+03	-0.107086D+01	0.958145D+01	0.808441D-01
46	0.0	0.374789D+03	0.0	0.709900D+01	0.807689D-01
47	0.277789D+02	0.393032D+03	-0.975681D+01	0.184080D+02	0.803328D-01
48	0.257263D+02	0.391295D+03	-0.774821D+01	0.178396D+02	0.803761D-01
49	0.236785D+02	0.388416D+03	-0.435339D+01	0.172524D+02	0.804467D-01
50	0.216506D+02	0.385536D+03	-0.505014D+01	0.166446D+02	0.805154D-01
51	0.196321D+02	0.384814D+03	-0.544011D+01	0.160137D+02	0.805324D-01
52	0.176226D+02	0.381493D+03	-0.515480D+01	0.153565D+02	0.806112D-01
53	0.156231D+02	0.381038D+03	-0.475397D+01	0.146707D+02	0.806220D-01
54	0.136335D+02	0.377121D+03	-0.395092D+01	0.139508D+02	0.807141D-01
55	0.116541D+02	0.377519D+03	-0.311115D+01	0.131917D+02	0.807097D-01
56	0.968564D+01	0.372383D+03	-0.221849D+01	0.123862D+02	0.808244D-01
57	0.772725D+01	0.374935D+03	-0.142233D+01	0.115244D+02	0.807677D-01
58	0.577723D+01	0.368008D+03	-0.385145D-01	0.105928D+02	0.809249D-01
59	0.383472D+01	0.368402D+03	0.149633D+01	0.957101D+01	0.809158D-01
60	0.192720D+01	0.359431D+03	-0.139946D+00	0.842615D+01	0.811184D-01
61	0.0	0.377407D+03	-0.559674D-13	0.709900D+01	0.807079D-01
62	0.277789D+02	0.389150D+03	-0.737534D+00	0.194025D+02	0.804296D-01
63	0.237119D+02	0.389020D+03	-0.372675D+00	0.172474D+02	0.804328D-01
64	0.196522D+02	0.387553D+03	0.111373D+01	0.160092D+02	0.804669D-01
65	0.156144D+02	0.385119D+03	0.348864D+01	0.146669D+02	0.805257D-01
66	0.116112D+02	0.381643D+03	0.663006D+01	0.131386D+02	0.806071D-01
67	0.765951D+01	0.376380D+03	0.994022D+01	0.115221D+02	0.807289D-01
68	0.379367D+01	0.378810D+03	0.116879D+02	0.956981D+01	0.806707D-01
69	0.0	0.358875D+03	0.613323D-01	0.709950D+01	0.811309D-01
70	0.277789D+02	0.393380D+03	-0.746689D+00	0.183970D+02	0.803129D-01
71	0.257216D+02	0.393575D+03	0.893952D+00	0.178291D+02	0.803227D-01
72	0.236629D+02	0.393935D+03	0.277521D+01	0.172424D+02	0.803137D-01
73	0.216066D+02	0.393263D+03	0.451027D+01	0.166351D+02	0.803297D-01
74	0.195491D+02	0.394325D+03	0.613907D+01	0.160048D+02	0.803033D-01
75	0.174945D+02	0.392103D+03	0.823032D+01	0.153486D+02	0.803563D-01
76	0.154453D+02	0.393103D+03	0.107059D+02	0.146630D+02	0.803331D-01
77	0.134029D+02	0.388658D+03	0.134133D+02	0.139438D+02	0.804359D-01
78	0.113723D+02	0.388872D+03	0.154344D+02	0.131854D+02	0.804280D-01
79	0.935883D+01	0.381019D+03	0.174455D+02	0.123807D+02	0.806115D-01
80	0.736830D+01	0.382078D+03	0.226876D+02	0.115199D+02	0.805823D-01

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
81	0.540672D+01	0.367249D+03	0.262556D+02	0.105893D+02	0.809208D-01
82	0.349044D+01	0.361666D+03	0.295454D+02	0.956862D+01	0.810413D-01
83	0.168667D+01	0.325666D+03	0.192820D+02	0.842522D+01	0.918268D-01
84	0.0	0.313948D+03	0.540917D-01	0.710000D+01	0.820725D-01
85	0.277789D+02	0.396752D+03	-0.127690D+01	0.133935D+02	0.802452D-01
86	0.236979D+02	0.398993D+03	0.269798D+00	0.172566D+02	0.801903D-01
87	0.196042D+02	0.399360D+03	0.648164D+01	0.160394D+02	0.801676D-01
88	0.155084D+02	0.398941D+03	0.147319D+02	0.147218D+02	0.801848D-01
89	0.114399D+02	0.395384D+03	0.251590D+02	0.132741D+02	0.802592D-01
90	0.743627D+01	0.386244D+03	0.382397D+02	0.116478D+02	0.804541D-01
91	0.357985D+01	0.373813D+03	0.558599D+02	0.975407D+01	0.806949D-01
92	0.0	0.331067D+03	0.892344D+02	0.739000D+01	0.814850D-01
93	0.277789D+02	0.296727D+03	-0.127682D+01	0.183900D+02	0.802458D-01
94	0.257725D+02	0.396925D+03	-0.190816D+01	0.178795D+02	0.802409D-01
95	0.237619D+02	0.406828D+03	-0.234275D+01	0.172715D+02	0.799956D-01
96	0.217139D+02	0.405376D+03	0.151249D+01	0.166842D+02	0.800315D-01
97	0.196675D+02	0.407428D+03	0.656449D+01	0.160754D+02	0.799795D-01
98	0.175163D+02	0.406118D+03	0.111751D+02	0.154426D+02	0.800097D-01
99	0.155657D+02	0.406902D+03	0.167520D+02	0.147828D+02	0.799853D-01
100	0.135226D+02	0.403345D+03	0.225935D+02	0.140921D+02	0.800669D-01
101	0.114878D+02	0.403504D+03	0.295768D+02	0.133658D+02	0.800518D-01
102	0.946999D+01	0.396113D+03	0.364711D+02	0.125976D+02	0.802199D-01
103	0.747377D+01	0.396475D+03	0.453566D+02	0.117795D+02	0.801886D-01
104	0.550695D+01	0.331996D+03	0.570017D+02	0.109901D+02	0.804957D-01
105	0.358596D+01	0.376867D+03	0.725295D+02	0.994331D+01	0.805578D-01
106	0.174924D+01	0.348509D+03	0.761643D+02	0.888403D+01	0.811784D-01
107	0.0	0.340029D+03	0.906744D+02	0.768000D+01	0.812846D-01
108	0.277789D+02	0.424285D+03	-0.137240D+02	0.132900D+02	0.795391D-01
109	0.237209D+02	0.425394D+03	-0.109719D+02	0.172285D+02	0.795173D-01
110	0.196655D+02	0.430738D+03	0.101157D+01	0.160971D+02	0.793307D-01
111	0.155992D+02	0.432139D+03	0.169955D+02	0.148200D+02	0.793348D-01
112	0.115479D+02	0.429297D+03	0.358324D+02	0.135540D+02	0.792792D-01
113	0.755357D+01	0.421738D+03	0.536729D+02	0.120834D+02	0.795104D-01
114	0.367674D+01	0.411491D+03	0.894960D+02	0.104070D+02	0.796323D-01
115	0.0	0.375501D+03	0.135049D+03	0.840250D+01	0.801820D-01
116	0.277789D+02	0.435182D+03	-0.192049D+02	0.181900D+02	0.792516D-01
117	0.258482D+02	0.442480D+03	-0.206144D+02	0.176972D+02	0.790539D-01
118	0.239390D+02	0.460200D+03	-0.215223D+02	0.171902D+02	0.785648D-01
119	0.218674D+02	0.467378D+03	-0.111619D+02	0.166678D+02	0.783560D-01
120	0.198130D+02	0.477268D+03	-0.505460D+01	0.161286D+02	0.780912D-01

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
121	0.177573D+02	0.476950D+03	0.613218D+01	0.155706D+02	0.781006D-01
122	0.156982D+02	0.478171D+03	0.184100D+02	0.149919D+02	0.780561D-01
123	0.136504D+02	0.474579D+03	0.304695D+02	0.143900D+02	0.781420D-01
124	0.116151D+02	0.473304D+03	0.438412D+02	0.137617D+02	0.781485D-01
125	0.960208D+01	0.466785D+03	0.568042D+02	0.131034D+02	0.782950D-01
126	0.761048D+01	0.465301D+03	0.721787D+02	0.124101D+02	0.782768D-01
127	0.564781D+01	0.455113D+03	0.877142D+02	0.116758D+02	0.784863D-01
128	0.371715D+01	0.454324D+03	0.108925D+03	0.108921D+02	0.783820D-01
129	0.183061D+01	0.435533D+03	0.125758D+03	0.100475D+02	0.787703D-01
130	0.0	0.430216D+03	0.154604D+03	0.912500D+01	0.766641D-01
131	0.277789D+02	0.459336D+03	-0.511880D+02	0.180305D+02	0.851827D-01
132	0.238384D+02	0.452776D+03	-0.268113D+02	0.170855D+02	0.847555D-01
133	0.197688D+02	0.471845D+03	-0.200776D+01	0.160850D+02	0.859688D-01
134	0.156199D+02	0.464273D+03	0.243234D+02	0.150190D+02	0.861083D-01
135	0.115716D+02	0.451673D+03	0.480709D+02	0.138672D+02	0.859806D-01
136	0.761289D+01	0.445420D+03	0.726277D+02	0.125162D+02	0.854599D-01
137	0.373931D+01	0.442517D+03	0.103154D+03	0.112241D+02	0.846881D-01
138	0.0	0.433578D+03	0.149797D+03	0.963300D+01	0.836965D-01
139	0.277789D+02	0.484689D+03	-0.331715D+02	0.178710D+02	0.911138D-01
140	0.257195D+02	0.464145D+03	-0.531874D+02	0.174329D+02	0.909070D-01
141	0.237324D+02	0.452668D+03	-0.322253D+02	0.169835D+02	0.912725D-01
142	0.217209D+02	0.455234D+03	-0.159548D+02	0.165219D+02	0.934009D-01
143	0.196337D+02	0.459300D+03	0.216737D+01	0.160470D+02	0.942876D-01
144	0.175490D+02	0.451705D+03	0.158932D+02	0.155576D+02	0.947018D-01
145	0.154797D+02	0.443874D+03	0.304305D+02	0.150523D+02	0.944583D-01
146	0.134618D+02	0.435676D+03	0.420655D+02	0.145295D+02	0.941465D-01
147	0.114805D+02	0.432220D+03	0.527213D+02	0.139871D+02	0.937028D-01
148	0.952291D+01	0.427784D+03	0.626723D+02	0.134229D+02	0.932515D-01
149	0.759098D+01	0.425152D+03	0.731461D+02	0.128338D+02	0.926466D-01
150	0.569146D+01	0.423751D+03	0.832118D+02	0.122164D+02	0.919957D-01
151	0.376472D+01	0.430802D+03	0.973581D+02	0.115660D+02	0.910172D-01
152	0.188737D+01	0.430157D+03	0.116828D+03	0.108769D+02	0.901053D-01
153	0.0	0.437539D+03	0.144990D+03	0.101410D+02	0.887288D-01
154	0.277789D+02	0.463475D+03	-0.654098D+01	0.178535D+02	0.919782D-01
155	0.237936D+02	0.478478D+03	0.218113D+01	0.165895D+02	0.904262D-01
156	0.196018D+02	0.469219D+03	0.156332D+02	0.160791D+02	0.946024D-01
157	0.154173D+02	0.457106D+03	0.262310D+02	0.151140D+02	0.940984D-01
158	0.113373D+02	0.441805D+03	0.429269D+02	0.140829D+02	0.935272D-01
159	0.748985D+01	0.435139D+03	0.605543D+02	0.129701D+02	0.925523D-01
160	0.367665D+01	0.429204D+03	0.311621D+02	0.117524D+02	0.914030D-01

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
161	0.0	0.417646D+03	0.101248D+03	0.103930D+02	0.900633D-01
162	0.277789D+02	0.478634D+03	-0.675492D+01	0.178360D+02	0.915078D-01
163	0.257762D+02	0.489155D+03	-0.421942D+00	0.174211D+02	0.902381D-01
164	0.237485D+02	0.509131D+03	0.109074D+02	0.169960D+02	0.894759D-01
165	0.216637D+02	0.494303D+03	0.147020D+02	0.165601D+02	0.922575D-01
166	0.195759D+02	0.488738D+03	0.160210D+02	0.161123D+02	0.934066D-01
167	0.174916D+02	0.436268D+03	0.200143D+02	0.156518D+02	0.936727D-01
168	0.154091D+02	0.480478D+03	0.248823D+02	0.151772D+02	0.934122D-01
169	0.133786D+02	0.471376D+03	0.301735D+02	0.146874D+02	0.932072D-01
170	0.113878D+02	0.466029D+03	0.359072D+02	0.141806D+02	0.928889D-01
171	0.942447D+01	0.460245D+03	0.437066D+02	0.136551D+02	0.925452D-01
172	0.748802D+01	0.458066D+03	0.526157D+02	0.131085D+02	0.920361D-01
173	0.557743D+01	0.451983D+03	0.602833D+02	0.125381D+02	0.916443D-01
174	0.369556D+01	0.462826D+03	0.707446D+02	0.119404D+02	0.906951D-01
175	0.181501D+01	0.446244D+03	0.833481D+02	0.113113D+02	0.904558D-01
176	0.0	0.435430D+03	0.105571D+03	0.106450D+02	0.899336D-01
177	0.277789D+02	0.510663D+03	-0.337745D+01	0.178360D+02	0.870875D-01
178	0.237504D+02	0.540534D+03	0.279675D+01	0.170166D+02	0.848954D-01
179	0.196144D+02	0.521489D+03	0.430486D+01	0.161557D+02	0.887179D-01
180	0.155076D+02	0.515943D+03	0.998054D+01	0.152462D+02	0.889123D-01
181	0.114764D+02	0.504370D+03	0.181038D+02	0.142789D+02	0.887375D-01
182	0.753848D+01	0.499503D+03	0.311005D+02	0.132412D+02	0.881460D-01
183	0.366660D+01	0.506208D+03	0.492025D+02	0.121149D+02	0.865557D-01
184	0.0	0.495835D+03	0.974175D+02	0.103735D+02	0.859131D-01
185	0.277789D+02	0.542652D+03	-0.554737D-12	0.178360D+02	0.926671D-01
186	0.258302D+02	0.550681D+03	-0.232564D+01	0.174413D+02	0.816462D-01
187	0.238734D+02	0.571703D+03	-0.519124D+01	0.170376D+02	0.803077D-01
188	0.218699D+02	0.559488D+03	-0.655618D+01	0.166240D+02	0.828558D-01
189	0.198451D+02	0.554124D+03	-0.751903D+01	0.161998D+02	0.839996D-01
190	0.178204D+02	0.551970D+03	-0.637672D+01	0.157642D+02	0.845056D-01
191	0.157911D+02	0.552209D+03	-0.490532D+01	0.153162D+02	0.843690D-01
192	0.137800D+02	0.544012D+03	-0.292914D+01	0.148548D+02	0.846665D-01
193	0.117395D+02	0.543322D+03	0.513807D-01	0.143786D+02	0.845058D-01
194	0.980828D+01	0.536337D+03	0.377482D+01	0.138860D+02	0.847922D-01
195	0.784318D+01	0.542303D+03	0.902227D+01	0.133753D+02	0.841797D-01
196	0.587684D+01	0.534651D+03	0.150310D+02	0.128443D+02	0.845342D-01
197	0.392691D+01	0.550699D+03	0.242730D+02	0.122904D+02	0.822020D-01
198	0.195719D+01	0.547950D+03	0.521546D+02	0.117104D+02	0.830268D-01
199	0.0	0.556139D+03	0.892649D+02	0.111030D+02	0.818927D-01
200	0.277789D+02	0.492100D+03	0.0	0.178360D+02	0.945902D-01

FINITE ELEMENT RESULTS

NODE	PSI(I)	VZ	VR	R(I)	DENSITY
201	0.236547D+02	0.520660D+03	0.528668D+01	0.170408D+02	0.935349D-01
202	0.193924D+02	0.495172D+03	0.105126D+02	0.162057D+02	0.972542D-01
203	0.152029D+02	0.491502D+03	0.144996D+02	0.153272D+02	0.977364D-01
204	0.111625D+02	0.463783D+03	0.164231D+02	0.143941D+02	0.977777D-01
205	0.725862D+01	0.450237D+03	0.163626D+02	0.133961D+02	0.977812D-01
206	0.346617D+01	0.423253D+03	0.132814D+02	0.123175D+02	0.962319D-01
207	0.0	0.388623D+03	0.337515D+01	0.111350D+02	0.938117D-01
208	0.277789D+02	0.503808D+03	0.0	0.178360D+02	0.942072D-01
209	0.256911D+02	0.520442D+03	0.0	0.174445D+02	0.931027D-01
210	0.235499D+02	0.542338D+03	0.0	0.170441D+02	0.929104D-01
211	0.213624D+02	0.520770D+03	0.0	0.166340D+02	0.956200D-01
212	0.192001D+02	0.510345D+03	0.0	0.162135D+02	0.967874D-01
213	0.170597D+02	0.499913D+03	0.0	0.157819D+02	0.973731D-01
214	0.149486D+02	0.489100D+03	0.0	0.153381D+02	0.974483D-01
215	0.128979D+02	0.473997D+03	0.0	0.148811D+02	0.976513D-01
216	0.109039D+02	0.463359D+03	0.0	0.144095D+02	0.977706D-01
217	0.895174D+01	0.450412D+03	0.0	0.139221D+02	0.979599D-01
218	0.704703D+01	0.443613D+03	0.0	0.134169D+02	0.979736D-01
219	0.518260D+01	0.426766D+03	0.0	0.128919D+02	0.982995D-01
220	0.338183D+01	0.426275D+03	0.0	0.123447D+02	0.981367D-01
221	0.162651D+01	0.392516D+03	0.0	0.117720D+02	0.998775D-01
222	0.0	0.362656D+03	0.0	0.111700D+02	0.994872D-01

NODE	WRL(I)	HT	VT	WT	HS
1	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
2	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
3	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
4	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
5	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
6	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
7	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
8	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
9	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
10	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
11	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
12	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
13	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
14	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
15	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
16	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
17	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
18	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
19	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
20	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
21	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
22	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
23	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
24	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
25	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
26	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
27	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
28	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
29	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
30	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
31	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
32	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
33	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
34	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
35	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
36	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
37	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
38	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
39	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
40	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03

NODE	WRL(I)	HT	VT	WT	HS
41	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
42	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
43	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
44	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
45	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
46	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
47	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
48	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
49	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
50	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
51	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
52	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
53	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
54	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
55	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
56	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
57	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
58	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
59	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
60	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
61	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
62	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
63	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
64	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
65	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
66	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
67	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
68	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
69	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
70	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
71	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
72	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
73	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
74	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
75	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
76	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
77	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
78	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
79	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03
80	0.0	0.1332000D+03	0.0	0.0	0.1313190D+03

NODE	WRL(1)	HT	VT	WT	HS
81	0.0	0.133200D+03	0.0	0.0	0.131319D+03
82	0.0	0.133200D+03	0.0	0.0	0.131319D+03
83	0.0	0.133200D+03	0.0	0.0	0.131319D+03
84	0.0	0.133200D+03	0.0	0.0	0.131319D+03
85	0.0	0.133200D+03	0.0	0.0	0.131319D+03
86	0.0	0.133200D+03	0.0	0.0	0.131319D+03
87	0.0	0.133200D+03	0.0	0.0	0.131319D+03
88	0.0	0.133200D+03	0.0	0.0	0.131319D+03
89	0.0	0.133200D+03	0.0	0.0	0.131319D+03
90	0.0	0.133200D+03	0.0	0.0	0.131319D+03
91	0.0	0.133200D+03	0.0	0.0	0.131319D+03
92	0.0	0.133200D+03	0.0	0.0	0.131319D+03
93	0.0	0.133200D+03	0.0	0.0	0.131319D+03
94	0.0	0.133200D+03	0.0	0.0	0.131319D+03
95	0.0	0.133200D+03	0.0	0.0	0.131319D+03
96	0.0	0.133200D+03	0.0	0.0	0.131319D+03
97	0.0	0.133200D+03	0.0	0.0	0.131319D+03
98	0.0	0.133200D+03	0.0	0.0	0.131319D+03
99	0.0	0.133200D+03	0.0	0.0	0.131319D+03
100	0.0	0.133200D+03	0.0	0.0	0.131319D+03
101	0.0	0.133200D+03	0.0	0.0	0.131319D+03
102	0.0	0.133200D+03	0.0	0.0	0.131319D+03
103	0.0	0.133200D+03	0.0	0.0	0.131319D+03
104	0.0	0.133200D+03	0.0	0.0	0.131319D+03
105	0.0	0.133200D+03	0.0	0.0	0.131319D+03
106	0.0	0.133200D+03	0.0	0.0	0.131319D+03
107	0.0	0.133200D+03	0.0	0.0	0.131319D+03
108	0.0	0.133200D+03	0.0	0.0	0.131319D+03
109	0.0	0.133200D+03	0.0	0.0	0.131319D+03
110	0.0	0.133200D+03	0.0	0.0	0.131319D+03
111	0.0	0.133200D+03	0.0	0.0	0.131319D+03
112	0.0	0.133200D+03	0.0	0.0	0.131319D+03
113	0.0	0.133200D+03	0.0	0.0	0.131319D+03
114	0.0	0.133200D+03	0.0	0.0	0.131319D+03
115	0.0	0.133200D+03	0.0	0.0	0.131319D+03
116	0.0	0.133200D+03	0.0	0.110723D+04	0.131319D+03
117	0.0	0.133200D+03	0.0	0.107723D+04	0.131319D+03
118	0.0	0.133200D+03	0.0	0.104637D+04	0.131319D+03
119	0.0	0.133200D+03	0.0	0.101457D+04	0.131319D+03
120	0.0	0.133200D+03	0.0	0.981747D+03	0.131319D+03

NODE	WRL(I)	HT	VT	WT	HS
121	0.0	0.133200D+03	0.0	0.947785D+03	0.131319D+03
122	0.0	0.133200D+03	0.0	0.912559D+03	0.131319D+03
123	0.0	0.133200D+03	0.0	0.875918D+03	0.131319D+03
124	0.0	0.133200D+03	0.0	0.837676D+03	0.131319D+03
125	0.0	0.133200D+03	0.0	0.797603D+03	0.131319D+03
126	0.0	0.133200D+03	0.0	0.755406D+03	0.131319D+03
127	0.0	0.133200D+03	0.0	0.710709D+03	0.131319D+03
128	0.0	0.133200D+03	0.0	0.663005D+03	0.131319D+03
129	0.0	0.133200D+03	0.0	0.611592D+03	0.131319D+03
130	0.0	0.133200D+03	0.0	0.555440D+03	0.131319D+03
131	0.225510D+03	0.140927D+03	0.150086D+03	0.883903D+03	0.135548D+03
132	0.248211D+03	0.141519D+03	0.174262D+03	0.802063D+03	0.135786D+03
133	0.216593D+03	0.140267D+03	0.161470D+03	0.769544D+03	0.134509D+03
134	0.213103D+03	0.140200D+03	0.170162D+03	0.692283D+03	0.134418D+03
135	0.210463D+03	0.140245D+03	0.181955D+03	0.600750D+03	0.134401D+03
136	0.199688D+03	0.140088D+03	0.189794D+03	0.504060D+03	0.134062D+03
137	0.186366D+03	0.139935D+03	0.199219D+03	0.392962D+03	0.133505D+03
138	0.166281D+03	0.139642D+03	0.207129D+03	0.276820D+03	0.132702D+03
139	0.529588D+03	0.148654D+03	0.355607D+03	0.732202D+03	0.141291D+03
140	0.545542D+03	0.149113D+03	0.375526D+03	0.685617D+03	0.141936D+03
141	0.552459D+03	0.149321D+03	0.390350D+03	0.643438D+03	0.142157D+03
142	0.521199D+03	0.148408D+03	0.373551D+03	0.627138D+03	0.141395D+03
143	0.495975D+03	0.147672D+03	0.370891D+03	0.605391D+03	0.140704D+03
144	0.492168D+03	0.147561D+03	0.379622D+03	0.567372D+03	0.140595D+03
145	0.488723D+03	0.147461D+03	0.380619D+03	0.526619D+03	0.140463D+03
146	0.482953D+03	0.147322D+03	0.399700D+03	0.484712D+03	0.140294D+03
147	0.479942D+03	0.147205D+03	0.411759D+03	0.439640D+03	0.140024D+03
148	0.476243D+03	0.147097D+03	0.425760D+03	0.391290D+03	0.139744D+03
149	0.472337D+03	0.146933D+03	0.441650D+03	0.339545D+03	0.139363D+03
150	0.468125D+03	0.146861D+03	0.459834D+03	0.283778D+03	0.138905D+03
151	0.462175D+03	0.146638D+03	0.479516D+03	0.224510D+03	0.138191D+03
152	0.451958D+03	0.146391D+03	0.498626D+03	0.163451D+03	0.137448D+03
153	0.441422D+03	0.146084D+03	0.522342D+03	0.949420D+02	0.136332D+03
154	0.529585D+03	0.148654D+03	0.355956D+03	0.0	0.141291D+03
155	0.552385D+03	0.149219D+03	0.390160D+03	0.0	0.142156D+03
156	0.495909D+03	0.147670D+03	0.370102D+03	0.0	0.140703D+03
157	0.488563D+03	0.147456D+03	0.387902D+03	0.0	0.140464D+03
158	0.479772D+03	0.147200D+03	0.408812D+03	0.0	0.140014D+03
159	0.472158D+03	0.146978D+03	0.436842D+03	0.0	0.139346D+03
160	0.461617D+03	0.146672D+03	0.471342D+03	0.0	0.136158D+03

NODE	WPL(I)	HT	VT	WT	HS
161	0.441422D+03	0.146084D+03	0.509677D+03	0.0	0.136382D+03
162	0.529589D+03	0.148654D+03	0.356305D+03	0.0	0.141291D+03
163	0.545250D+03	0.149110D+03	0.375579D+03	0.0	0.141925D+03
164	0.552450D+03	0.149320D+03	0.390056D+03	0.0	0.142157D+03
165	0.520390D+03	0.148385D+03	0.377093D+03	0.0	0.141374D+03
166	0.495858D+03	0.147669D+03	0.369301D+03	0.0	0.140701D+03
167	0.492066D+03	0.147558D+03	0.377261D+03	0.0	0.140591D+03
168	0.488540D+03	0.147455D+03	0.386268D+03	0.0	0.140463D+03
169	0.483769D+03	0.147316D+03	0.395253D+03	0.0	0.140284D+03
170	0.479772D+03	0.147200D+03	0.405995D+03	0.0	0.140014D+03
171	0.476053D+03	0.147092D+03	0.413352D+03	0.0	0.139728D+03
172	0.472154D+03	0.146978D+03	0.432228D+03	0.0	0.139345D+03
173	0.467857D+03	0.146853D+03	0.447779D+03	0.0	0.138874D+03
174	0.461715D+03	0.146675D+03	0.464018D+03	0.0	0.138164D+03
175	0.451580D+03	0.146380D+03	0.479075D+03	0.0	0.137414D+03
176	0.441422D+03	0.146084D+03	0.497611D+03	0.0	0.136382D+03
177	0.220015D+03	0.148654D+03	0.148025D+03	0.0	0.141291D+03
178	0.245775D+03	0.149320D+03	0.173357D+03	0.0	0.142157D+03
179	0.228362D+03	0.147680D+03	0.169557D+03	0.0	0.140712D+03
180	0.224901D+03	0.147460D+03	0.176939D+03	0.0	0.140465D+03
181	0.222193D+03	0.147204D+03	0.186397D+03	0.0	0.140027D+03
182	0.217903D+03	0.146981D+03	0.196639D+03	0.0	0.139356D+03
183	0.211458D+03	0.146670D+03	0.207670D+03	0.0	0.138155D+03
184	0.199753D+03	0.146084D+03	0.214869D+03	0.0	0.136382D+03
185	0.849479D+03	0.148654D+03	0.571527D+03	0.0	0.141291D+03
186	0.842917D+03	0.149101D+03	0.579944D+03	0.0	0.141913D+03
187	0.854864D+03	0.149315D+03	0.602103D+03	0.0	0.142154D+03
188	0.816313D+03	0.148465D+03	0.559255D+03	0.0	0.141444D+03
189	0.786303D+03	0.147746D+03	0.563935D+03	0.0	0.140779D+03
190	0.763692D+03	0.147575D+03	0.581335D+03	0.0	0.140609D+03
191	0.742293D+03	0.147473D+03	0.561571D+03	0.0	0.140488D+03
192	0.709226D+03	0.147341D+03	0.572925D+03	0.0	0.140326D+03
193	0.685507D+03	0.147221D+03	0.572182D+03	0.0	0.140074D+03
194	0.653685D+03	0.147113D+03	0.564900D+03	0.0	0.139789D+03
195	0.636660D+03	0.146999D+03	0.571195D+03	0.0	0.139420D+03
196	0.602913D+03	0.146676D+03	0.563280D+03	0.0	0.138960D+03
197	0.594572D+03	0.146695D+03	0.580522D+03	0.0	0.138261D+03
198	0.565679D+03	0.146402D+03	0.579671D+03	0.0	0.137479D+03
199	0.548743D+03	0.146084D+03	0.593236D+03	0.0	0.136382D+03
200	-0.611177D+02	0.148654D+03	-0.411198D+02	0.0	0.141291D+03

NODE	WRL(I)	HT	VT	WT	HS
201	-0.261775E+02	0.149215D+03	-0.184340D+02	0.0	0.142074D+03
202	-0.131787E+02	0.147702D+03	-0.975800D+01	0.0	0.140738D+03
203	-0.106428E+02	0.147434D+03	-0.833249D+01	0.0	0.140450D+03
204	-0.670200E+01	0.147188D+03	-0.558731D+01	0.0	0.139995D+03
205	-0.221630E+01	0.146969D+03	-0.199532D+01	0.0	0.139311D+03
206	-0.382442E+01	0.146636D+03	-0.372583D+01	0.0	0.138118D+03
207	-0.178013E+02	0.146084D+03	-0.191841D+02	0.0	0.136382D+03
208	-0.611177E+02	0.148654D+03	-0.411198D+02	0.0	0.141291D+03
209	-0.369251E+02	0.149124D+03	-0.254005D+02	0.0	0.141943D+03
210	-0.255710E+02	0.149169D+03	-0.180034D+02	0.0	0.142037D+03
211	-0.163603E+02	0.148274D+03	-0.119025D+02	0.0	0.141275D+03
212	-0.129754E+02	0.147685D+03	-0.960336D+01	0.0	0.140721D+03
213	-0.113191E+02	0.147529D+03	-0.860665D+01	0.0	0.140559D+03
214	-0.105064E+02	0.147417D+03	-0.821987D+01	0.0	0.140432D+03
215	-0.844756E+01	0.147233D+03	-0.631214D+01	0.0	0.140227D+03
216	-0.666134E+01	0.147174D+03	-0.554744D+01	0.0	0.139960D+03
217	-0.507688E+01	0.147065D+03	-0.437597D+01	0.0	0.139642D+03
218	-0.196173E+01	0.146957D+03	-0.175456D+01	0.0	0.139267D+03
219	-0.162425E+01	0.146820D+03	-0.151188D+01	0.0	0.138745D+03
220	-0.398225E+01	0.146621D+03	-0.387106D+01	0.0	0.138082D+03
221	-0.909343E+01	0.146351D+03	-0.926955D+01	0.0	0.137321D+03
222	-0.176013E+02	0.146084D+03	-0.191240D+02	0.0	0.136392D+03

NODE	TEMP	TTOT	PRESS	PTOT	RHOT
1	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
2	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
3	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
4	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
5	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
6	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
7	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
8	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
9	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
10	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
11	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
12	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
13	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
14	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
15	0.547162D+03	0.555000D+03	0.166501D+02	0.175000D+02	0.851884D-01
16	0.543178D+03	0.555000D+03	0.162297D+02	0.175000D+02	0.851884D-01
17	0.543390D+03	0.555000D+03	0.162519D+02	0.175000D+02	0.851884D-01
18	0.543686D+03	0.555000D+03	0.162829D+02	0.175000D+02	0.851884D-01
19	0.543879D+03	0.555000D+03	0.163032D+02	0.175000D+02	0.851884D-01
20	0.543987D+03	0.555000D+03	0.163145D+02	0.175000D+02	0.851884D-01
21	0.543990D+03	0.555000D+03	0.163148D+02	0.175000D+02	0.851884D-01
22	0.543719D+03	0.555000D+03	0.162863D+02	0.175000D+02	0.851884D-01
23	0.543774D+03	0.555000D+03	0.162921D+02	0.175000D+02	0.851884D-01
24	0.541022D+03	0.555000D+03	0.160054D+02	0.175000D+02	0.851884D-01
25	0.541644D+03	0.555000D+03	0.160698D+02	0.175000D+02	0.851884D-01
26	0.542556D+03	0.555000D+03	0.161647D+02	0.175000D+02	0.851884D-01
27	0.542836D+03	0.555000D+03	0.162002D+02	0.175000D+02	0.851884D-01
28	0.543057D+03	0.555000D+03	0.162170D+02	0.175000D+02	0.851884D-01
29	0.543321D+03	0.555000D+03	0.162447D+02	0.175000D+02	0.851884D-01
30	0.543369D+03	0.555000D+03	0.162497D+02	0.175000D+02	0.851884D-01
31	0.543614D+03	0.555000D+03	0.162753D+02	0.175000D+02	0.851884D-01
32	0.543571D+03	0.555000D+03	0.162709D+02	0.175000D+02	0.851884D-01
33	0.543850D+03	0.555000D+03	0.163001D+02	0.175000D+02	0.851884D-01
34	0.543649D+03	0.555000D+03	0.162790D+02	0.175000D+02	0.851884D-01
35	0.544076D+03	0.555000D+03	0.163239D+02	0.175000D+02	0.851884D-01
36	0.543430D+03	0.555000D+03	0.162561D+02	0.175000D+02	0.851884D-01
37	0.544333D+03	0.555000D+03	0.163508D+02	0.175000D+02	0.851884D-01
38	0.543584D+03	0.555000D+03	0.162722D+02	0.175000D+02	0.851884D-01
39	0.542108D+03	0.555000D+03	0.161181D+02	0.175000D+02	0.851884D-01
40	0.542397D+03	0.555000D+03	0.161482D+02	0.175000D+02	0.851884D-01

NCDE	TEMP	TTOT	PRESS	PTOT	RHCT
41	0.542793D+03	0.555000D+03	0.161895D+02	0.175000D+02	0.851884D-01
42	0.543094D+03	0.555000D+03	0.162209D+02	0.175000D+02	0.851884D-01
43	0.543317D+03	0.555000D+03	0.162443D+02	0.175000D+02	0.851884D-01
44	0.543458D+03	0.555000D+03	0.162590D+02	0.175000D+02	0.851884D-01
45	0.543501D+03	0.555000D+03	0.162635D+02	0.175000D+02	0.851884D-01
46	0.543298D+03	0.555000D+03	0.162423D+02	0.175000D+02	0.851884D-01
47	0.542123D+03	0.555000D+03	0.161197D+02	0.175000D+02	0.851884D-01
48	0.542240D+03	0.555000D+03	0.161318D+02	0.175000D+02	0.851884D-01
49	0.542430D+03	0.555000D+03	0.161517D+02	0.175000D+02	0.851884D-01
50	0.542616D+03	0.555000D+03	0.161710D+02	0.175000D+02	0.851884D-01
51	0.542662D+03	0.555000D+03	0.161758D+02	0.175000D+02	0.851884D-01
52	0.542874D+03	0.555000D+03	0.161979D+02	0.175000D+02	0.851884D-01
53	0.542903D+03	0.555000D+03	0.162010D+02	0.175000D+02	0.851884D-01
54	0.543151D+03	0.555000D+03	0.162269D+02	0.175000D+02	0.851884D-01
55	0.543139D+03	0.555000D+03	0.162254D+02	0.175000D+02	0.851884D-01
56	0.543448D+03	0.555000D+03	0.162579D+02	0.175000D+02	0.851884D-01
57	0.543295D+03	0.555000D+03	0.162420D+02	0.175000D+02	0.851884D-01
58	0.543718D+03	0.555000D+03	0.162862D+02	0.175000D+02	0.851884D-01
59	0.543694D+03	0.555000D+03	0.162837D+02	0.175000D+02	0.851884D-01
60	0.544238D+03	0.555000D+03	0.163408D+02	0.175000D+02	0.851884D-01
61	0.543134D+03	0.555000D+03	0.162251D+02	0.175000D+02	0.851884D-01
62	0.542384D+03	0.555000D+03	0.161469D+02	0.175000D+02	0.851884D-01
63	0.542393D+03	0.555000D+03	0.161477D+02	0.175000D+02	0.851884D-01
64	0.542485D+03	0.555000D+03	0.161573D+02	0.175000D+02	0.851884D-01
65	0.542643D+03	0.555000D+03	0.161739D+02	0.175000D+02	0.851884D-01
66	0.542863D+03	0.555000D+03	0.161968D+02	0.175000D+02	0.851884D-01
67	0.542191D+03	0.555000D+03	0.162310D+02	0.175000D+02	0.851884D-01
68	0.543034D+03	0.555000D+03	0.162147D+02	0.175000D+02	0.851884D-01
69	0.544271D+03	0.555000D+03	0.163443D+02	0.175000D+02	0.851884D-01
70	0.542069D+03	0.555000D+03	0.161140D+02	0.175000D+02	0.851884D-01
71	0.542096D+03	0.555000D+03	0.161169D+02	0.175000D+02	0.851884D-01
72	0.542072D+03	0.555000D+03	0.161143D+02	0.175000D+02	0.851884D-01
73	0.542115D+03	0.555000D+03	0.161188D+02	0.175000D+02	0.851884D-01
74	0.542044D+03	0.555000D+03	0.161114D+02	0.175000D+02	0.851884D-01
75	0.542187D+03	0.555000D+03	0.161262D+02	0.175000D+02	0.851884D-01
76	0.542124D+03	0.555000D+03	0.161197D+02	0.175000D+02	0.851884D-01
77	0.542401D+03	0.555000D+03	0.161486D+02	0.175000D+02	0.851884D-01
78	0.542380D+03	0.555000D+03	0.161464D+02	0.175000D+02	0.851884D-01
79	0.542875D+03	0.555000D+03	0.161980D+02	0.175000D+02	0.851884D-01
80	0.542796D+03	0.555000D+03	0.161898D+02	0.175000D+02	0.851884D-01

NODE	TEMP	TTOT	PRESS	PTOT	RHOT
81	0.543707D+03	0.555000D+03	0.162851D+02	0.175000D+02	0.851884D-01
82	0.544031D+03	0.555000D+03	0.163190D+02	0.175000D+02	0.851884D-01
83	0.546134D+03	0.555000D+03	0.165409D+02	0.175000D+02	0.851884D-01
84	0.546789D+03	0.555000D+03	0.166105D+02	0.175000D+02	0.851884D-01
85	0.541887D+03	0.555000D+03	0.160950D+02	0.175000D+02	0.851884D-01
86	0.541738D+03	0.555000D+03	0.160796D+02	0.175000D+02	0.851884D-01
87	0.541677D+03	0.555000D+03	0.160733D+02	0.175000D+02	0.851884D-01
88	0.541724D+03	0.555000D+03	0.160781D+02	0.175000D+02	0.851884D-01
89	0.541924D+03	0.555000D+03	0.160990D+02	0.175000D+02	0.851884D-01
90	0.542450D+03	0.555000D+03	0.161537D+02	0.175000D+02	0.851884D-01
91	0.543099D+03	0.555000D+03	0.162215D+02	0.175000D+02	0.851884D-01
92	0.545220D+03	0.555000D+03	0.164443D+02	0.175000D+02	0.851884D-01
93	0.541888D+03	0.555000D+03	0.160957D+02	0.175000D+02	0.851884D-01
94	0.541875D+03	0.555000D+03	0.160938D+02	0.175000D+02	0.851884D-01
95	0.541212D+03	0.555000D+03	0.160250D+02	0.175000D+02	0.851884D-01
96	0.541309D+03	0.555000D+03	0.160351D+02	0.175000D+02	0.851884D-01
97	0.541168D+03	0.555000D+03	0.160205D+02	0.175000D+02	0.851884D-01
98	0.541250D+03	0.555000D+03	0.160290D+02	0.175000D+02	0.851884D-01
99	0.541184D+03	0.555000D+03	0.160221D+02	0.175000D+02	0.851884D-01
100	0.541405D+03	0.555000D+03	0.160450D+02	0.175000D+02	0.851884D-01
101	0.541364D+03	0.555000D+03	0.160403D+02	0.175000D+02	0.851884D-01
102	0.541818D+03	0.555000D+03	0.160879D+02	0.175000D+02	0.851884D-01
103	0.541734D+03	0.555000D+03	0.160792D+02	0.175000D+02	0.851884D-01
104	0.542573D+03	0.555000D+03	0.161666D+02	0.175000D+02	0.851884D-01
105	0.542730D+03	0.555000D+03	0.161829D+02	0.175000D+02	0.851884D-01
106	0.544399D+03	0.555000D+03	0.163577D+02	0.175000D+02	0.851884D-01
107	0.544683D+03	0.555000D+03	0.163877D+02	0.175000D+02	0.851884D-01
108	0.539974D+03	0.555000D+03	0.159971D+02	0.175000D+02	0.851884D-01
109	0.539915D+03	0.555000D+03	0.159910D+02	0.175000D+02	0.851884D-01
110	0.539544D+03	0.555000D+03	0.153528D+02	0.175000D+02	0.851884D-01
111	0.539419D+03	0.555000D+03	0.153400D+02	0.175000D+02	0.851884D-01
112	0.539540D+03	0.555000D+03	0.158524D+02	0.175000D+02	0.851884D-01
113	0.539896D+03	0.555000D+03	0.158891D+02	0.175000D+02	0.851884D-01
114	0.540227D+03	0.555000D+03	0.159222D+02	0.175000D+02	0.851884D-01
115	0.541716D+03	0.555000D+03	0.160773D+02	0.175000D+02	0.851884D-01
116	0.539193D+03	0.555000D+03	0.158167D+02	0.175000D+02	0.851884D-01
117	0.538654D+03	0.555000D+03	0.157615D+02	0.175000D+02	0.851884D-01
118	0.537319D+03	0.555000D+03	0.156252D+02	0.175000D+02	0.851884D-01
119	0.536747D+03	0.555000D+03	0.155671D+02	0.175000D+02	0.851884D-01
120	0.536021D+03	0.555000D+03	0.154935D+02	0.175000D+02	0.851884D-01

NODE	TEMP	TTOT	PRESS	PTOT	PFCT
121	0.536047D+03	0.555000D+03	0.154961D+02	0.175000D+02	0.851884D-01
122	0.535924D+03	0.555000D+03	0.154837D+02	0.175000D+02	0.851884D-01
123	0.536160D+03	0.555000D+03	0.155076D+02	0.175000D+02	0.851884D-01
124	0.536178D+03	0.555000D+03	0.155094D+02	0.175000D+02	0.851884D-01
125	0.536580D+03	0.555000D+03	0.155501D+02	0.175000D+02	0.851884D-01
126	0.536530D+03	0.555000D+03	0.155450D+02	0.175000D+02	0.851884D-01
127	0.537104D+03	0.555000D+03	0.156033D+02	0.175000D+02	0.851884D-01
128	0.536818D+03	0.555000D+03	0.155743D+02	0.175000D+02	0.851884D-01
129	0.537881D+03	0.555000D+03	0.156824D+02	0.175000D+02	0.851884D-01
130	0.537590D+03	0.555000D+03	0.156528D+02	0.175000D+02	0.851884D-01
131	0.562952D+03	0.587196D+03	0.178354D+02	0.206089D+02	0.943206D-01
132	0.565775D+03	0.589662D+03	0.178144D+02	0.206396D+02	0.940448D-01
133	0.560452D+03	0.584448D+03	0.179050D+02	0.207840D+02	0.955544D-01
134	0.560077D+03	0.584166D+03	0.179227D+02	0.208194D+02	0.957589D-01
135	0.560050D+03	0.584354D+03	0.178910D+02	0.208211D+02	0.957340D-01
136	0.558593D+03	0.583699D+03	0.177281D+02	0.207442D+02	0.955084D-01
137	0.556271D+03	0.583060D+03	0.174824D+02	0.206970D+02	0.954104D-01
138	0.552923D+03	0.581842D+03	0.171578D+02	0.206189D+02	0.952753D-01
139	0.588711D+03	0.619392D+03	0.198542D+02	0.237177D+02	0.103453D+00
140	0.591400D+03	0.621329D+03	0.198995D+02	0.236530D+02	0.102849D+00
141	0.592320D+03	0.622169D+03	0.199887D+02	0.237421D+02	0.103097D+00
142	0.589145D+03	0.618368D+03	0.203675D+02	0.241282D+02	0.105418D+00
143	0.586267D+03	0.615301D+03	0.204604D+02	0.242318D+02	0.106398D+00
144	0.585811D+03	0.614838D+03	0.205343D+02	0.243215D+02	0.106872D+00
145	0.585283D+03	0.614419D+03	0.204631D+02	0.242559D+02	0.106656D+00
146	0.584557D+03	0.613840D+03	0.203702D+02	0.241710D+02	0.106384D+00
147	0.583435D+03	0.613353D+03	0.202353D+02	0.241059D+02	0.106181D+00
148	0.582268D+03	0.612905D+03	0.200976D+02	0.240456D+02	0.106006D+00
149	0.580679D+03	0.612431D+03	0.199127D+02	0.239914D+02	0.105836D+00
150	0.578771D+03	0.611921D+03	0.197078D+02	0.239497D+02	0.105740D+00
151	0.575795D+03	0.611200D+03	0.193980D+02	0.239034D+02	0.105660D+00
152	0.572699D+03	0.609962D+03	0.191004D+02	0.238155D+02	0.105485D+00
153	0.568256D+03	0.609685D+03	0.186627D+02	0.237379D+02	0.105362D+00
154	0.590939D+03	0.619392D+03	0.201183D+02	0.237177D+02	0.103453D+00
155	0.590408D+03	0.622161D+03	0.197611D+02	0.237377D+02	0.103079D+00
156	0.585521D+03	0.615293D+03	0.203726D+02	0.242345D+02	0.106411D+00
157	0.584393D+03	0.614400D+03	0.203541D+02	0.242529D+02	0.106647D+00
158	0.582996D+03	0.613333D+03	0.201822D+02	0.241032D+02	0.106173D+00
159	0.580434D+03	0.612410D+03	0.198841D+02	0.239893D+02	0.105821D+00
160	0.576731D+03	0.611133D+03	0.195119D+02	0.238983D+02	0.105649D+00

NODE	TEMP	TTOT	PRESS	PTOT	RHOT
161	0.571660D+03	0.608685D+03	0.190568D+02	0.237379D+02	0.105362D+00
162	0.589728D+03	0.619392D+03	0.199745D+02	0.237177D+02	0.103453D+00
163	0.589610D+03	0.621293D+03	0.196933D+02	0.236527D+02	0.102854D+00
164	0.587890D+03	0.622166D+03	0.194701D+02	0.237415D+02	0.103094D+00
165	0.586051D+03	0.618269D+03	0.200125D+02	0.241351D+02	0.105465D+00
166	0.584005D+03	0.615287D+03	0.201911D+02	0.242366D+02	0.106422D+00
167	0.583238D+03	0.614825D+03	0.202220D+02	0.243218D+02	0.105875D+00
168	0.582695D+03	0.614397D+03	0.201466D+02	0.242525D+02	0.106646D+00
169	0.582217D+03	0.613817D+03	0.200863D+02	0.241679D+02	0.106374D+00
170	0.581401D+03	0.613333D+03	0.199896D+02	0.241032D+02	0.106173D+00
171	0.580496D+03	0.612881D+03	0.198847D+02	0.240457D+02	0.105998D+00
172	0.579137D+03	0.612410D+03	0.197289D+02	0.239893D+02	0.105830D+00
173	0.577864D+03	0.611988D+03	0.196018D+02	0.239474D+02	0.105736D+00
174	0.574047D+03	0.611145D+03	0.193008D+02	0.238992D+02	0.105651D+00
175	0.573557D+03	0.609916D+03	0.192034D+02	0.238124D+02	0.105480D+00
176	0.571330D+03	0.608685D+03	0.190184D+02	0.237379D+02	0.105362D+00
177	0.578688D+03	0.619392D+03	0.186717D+02	0.236445D+02	0.103133D+00
178	0.576289D+03	0.622158D+03	0.181285D+02	0.236545D+02	0.102717D+00
179	0.572700D+03	0.615332D+03	0.188260D+02	0.241576D+02	0.106067D+00
180	0.571877D+03	0.614415D+03	0.188386D+02	0.241724D+02	0.106290D+00
181	0.571592D+03	0.613352D+03	0.187892D+02	0.240119D+02	0.105767D+00
182	0.570086D+03	0.612422D+03	0.186129D+02	0.238841D+02	0.105364D+00
183	0.566402D+03	0.611127D+03	0.182418D+02	0.237705D+02	0.105086D+00
184	0.562132D+03	0.608685D+03	0.178994D+02	0.235982D+02	0.104742D+00
185	0.567647D+03	0.619392D+03	0.173690D+02	0.235712D+02	0.102814D+00
186	0.567974D+03	0.621255D+03	0.171644D+02	0.234972D+02	0.102162D+00
187	0.564714D+03	0.622145D+03	0.167861D+02	0.235597D+02	0.102309D+00
188	0.563597D+03	0.618603D+03	0.172345D+02	0.239446D+02	0.104576D+00
189	0.561592D+03	0.615609D+03	0.174607D+02	0.240800D+02	0.105678D+00
190	0.561357D+03	0.614894D+03	0.175586D+02	0.241520D+02	0.106118D+00
191	0.560391D+03	0.614471D+03	0.175156D+02	0.241050D+02	0.105984D+00
192	0.561923D+03	0.613922D+03	0.176098D+02	0.240038D+02	0.105634D+00
193	0.561555D+03	0.613421D+03	0.175648D+02	0.239290D+02	0.105391D+00
194	0.562416D+03	0.612969D+03	0.175514D+02	0.238567D+02	0.105149D+00
195	0.560310D+03	0.612295D+03	0.174738D+02	0.237899D+02	0.104936D+00
196	0.561719D+03	0.611982D+03	0.175758D+02	0.237238D+02	0.104732D+00
197	0.557858D+03	0.611245D+03	0.171802D+02	0.236564D+02	0.104561D+00
198	0.556776D+03	0.610007D+03	0.171105D+02	0.235534D+02	0.104317D+00
199	0.552933D+03	0.608685D+03	0.167603D+02	0.234535D+02	0.104122D+00
200	0.599078D+03	0.619392D+03	0.209746D+02	0.235712D+02	0.102314D+00

NODE	TEMP	TTOT	PRESS	PTOT	RHOT
201	0.599118D+03	0.621731D+03	0.207420D+02	0.236139D+02	0.102512D+00
202	0.594983D+03	0.615426D+03	0.214179D+02	0.241061D+02	0.105825D+00
203	0.594971D+03	0.614308D+03	0.215237D+02	0.240732D+02	0.105872D+00
204	0.595339D+03	0.613283D+03	0.215461D+02	0.239059D+02	0.105313D+00
205	0.595455D+03	0.612369D+03	0.215511D+02	0.237708D+02	0.104874D+00
206	0.596043D+03	0.610983D+03	0.216718D+02	0.236333D+02	0.104503D+00
207	0.596071D+03	0.608685D+03	0.218008D+02	0.234585D+02	0.104122D+00
208	0.598107D+03	0.619392D+03	0.208558D+02	0.235712D+02	0.102814D+00
209	0.598734D+03	0.621351D+03	0.206329D+02	0.234922D+02	0.102146D+00
210	0.597010D+03	0.621539D+03	0.205310D+02	0.236383D+02	0.102750D+00
211	0.595205D+03	0.617809D+03	0.210659D+02	0.240014D+02	0.104959D+00
212	0.593648D+03	0.615353D+03	0.212673D+02	0.241154D+02	0.105878D+00
213	0.593878D+03	0.614704D+03	0.214043D+02	0.241485D+02	0.106135D+00
214	0.594306D+03	0.614239D+03	0.214364D+02	0.240601D+02	0.105827D+00
215	0.594978D+03	0.613698D+03	0.215052D+02	0.239690D+02	0.105515D+00
216	0.595337D+03	0.613225D+03	0.215445D+02	0.238366D+02	0.105281D+00
217	0.595868D+03	0.612759D+03	0.216054D+02	0.238275D+02	0.105055D+00
218	0.595424D+03	0.612319D+03	0.216105D+02	0.237639D+02	0.104852D+00
219	0.596579D+03	0.611752D+03	0.217062D+02	0.237006D+02	0.104669D+00
220	0.595781D+03	0.610920D+03	0.216413D+02	0.236278D+02	0.104490D+00
221	0.596954D+03	0.609795D+03	0.218475D+02	0.235372D+02	0.104281D+00
222	0.597698D+03	0.608685D+03	0.220097D+02	0.234535D+02	0.104122D+00

I	ALPHA	BETA
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0
11	0.0	0.0
12	0.0	0.0
13	0.0	0.0
14	0.0	0.0
15	0.0	0.0
16	0.0	0.0
17	0.0	0.0
18	0.0	0.0
19	0.0	0.0
20	0.0	0.0
21	0.0	0.0
22	0.0	0.0
23	0.0	0.0
24	0.0	0.0
25	0.0	0.0
26	0.0	0.0
27	0.0	0.0
28	0.0	0.0
29	0.0	0.0
30	0.0	0.0
31	0.0	0.0
32	0.0	0.0
33	0.0	0.0
34	0.0	0.0
35	0.0	0.0
36	0.0	0.0
37	0.0	0.0
38	0.0	0.0
39	0.0	0.0
40	0.0	0.0
41	0.0	0.0
42	0.0	0.0
43	0.0	0.0
44	0.0	0.0
45	0.0	0.0
46	0.0	0.0
47	0.0	0.0
48	0.0	0.0
49	0.0	0.0
50	0.0	0.0
51	0.0	0.0
52	0.0	0.0
53	0.0	0.0
54	0.0	0.0
55	0.0	0.0
56	0.0	0.0
57	0.0	0.0
58	0.0	0.0
59	0.0	0.0
60	0.0	0.0
61	0.0	0.0
62	0.0	0.0
63	0.0	0.0
64	0.0	0.0
65	0.0	0.0
66	0.0	0.0
67	0.0	0.0
68	0.0	0.0
69	0.0	0.0
70	0.0	0.0
71	0.0	0.0
72	0.0	0.0
73	0.0	0.0
74	0.0	0.0
75	0.0	0.0
76	0.0	0.0
77	0.0	0.0
78	0.0	0.0
79	0.0	0.0
80	0.0	0.0
81	0.0	0.0
82	0.0	0.0
83	0.0	0.0
84	0.0	0.0
85	0.0	0.0
86	0.0	0.0
87	0.0	0.0
88	0.0	0.0
89	0.0	0.0
90	0.0	0.0
91	0.0	0.0
92	0.0	0.0
93	0.0	0.0
94	0.0	0.0
95	0.0	0.0
96	0.0	0.0
97	0.0	0.0
98	0.0	0.0
99	0.0	0.0
100	0.0	0.0
101	0.0	0.0
102	0.0	0.0
103	0.0	0.0
104	0.0	0.0
105	0.0	0.0
106	0.0	0.0
107	0.0	0.0
108	0.0	0.0
109	0.0	0.0
110	0.0	0.0
111	0.0	0.0
112	0.0	0.0
113	0.0	0.0
114	0.0	0.0
115	0.0	0.0
116	0.0	68.524267
117	0.0	67.647352
118	0.0	66.236705
119	0.0	65.232862
120	0.0	64.071488
121	0.0	63.285336
122	0.0	62.328517
123	0.0	61.501447
124	0.0	60.427729
125	0.0	59.478407
126	0.0	58.063552
127	0.0	56.889345
128	0.0	54.830135
129	0.0	52.453335
130	0.0	50.544037
131	17.935587	62.318772
132	21.015261	60.509622
133	18.891043	58.484825
134	20.101512	56.113930
135	21.828385	52.904708
136	22.808330	48.159885
137	23.674282	40.853784
138	24.286757	31.091447
139	35.871161	56.113286
140	38.792641	55.729302
141	40.703648	54.804893
142	39.728045	54.007690
143	38.921092	52.835506
144	40.024677	51.455952
145	41.209103	49.807077
146	42.380643	47.904548
147	43.590332	45.276052
148	44.592794	42.178706
149	45.672794	38.205559
150	46.797303	33.309604

151	47.352973	26.945351
152	46.205312	20.137770
153	48.573445	11.638799
154	37.522090	0.0
155	39.194172	0.0
156	38.249535	0.0
157	40.264267	0.0
158	42.644644	0.0
159	44.837178	0.0
160	47.177568	0.0
161	49.863456	0.0
162	36.662038	0.0
163	37.517501	0.0
164	37.450088	0.0
165	37.326953	0.0
166	37.060664	0.0
167	37.781893	0.0
168	38.759196	0.0
169	39.922428	0.0
170	40.977704	0.0
171	42.142144	0.0
172	43.150234	0.0
173	44.479770	0.0
174	44.742891	0.0
175	46.482382	0.0
176	47.997274	0.0
177	16.164506	0.0
178	17.779586	0.0
179	18.016310	0.0
180	18.917928	0.0
181	20.267660	0.0
182	21.428172	0.0
183	22.185938	0.0
184	23.020279	0.0
185	-4.333047	0.0
186	-2.724489	0.0
187	-1.942182	0.0
188	-1.321081	0.0
189	-1.048351	0.0
190	-0.929789	0.0
191	-0.882696	0.0
192	-0.810042	0.0
193	-0.583789	0.0
194	-0.556746	0.0
195	-0.299997	0.0
196	-0.135359	0.0
197	-0.339125	0.0
198	-0.834692	0.0
199	-1.956694	0.0
200	-4.776527	0.0
201	-2.027602	0.0
202	-1.128587	0.0
203	-0.990967	0.0
204	-0.689791	0.0
205	-0.252444	0.0
206	-0.504105	0.0
207	-2.825940	0.0
208	-4.666014	0.0
209	-2.794146	0.0
210	-1.901285	0.0
211	-1.298307	0.0
212	-1.078031	0.0
213	-0.986325	0.0
214	-0.962829	0.0
215	-0.823380	0.0
216	-0.685926	0.0
217	-0.556639	0.0
218	-0.226610	0.0
219	-0.202978	0.0
220	-0.520297	0.0
221	-1.352831	0.0
222	-3.018592	0.0

```
>DO YOU WISH TO ENTER THE PLOT SEQUENCE?
  1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.

>DO YOU WISH TO CONTINUE PLOT SEQUENCE?
    1 = YES; 2 = NO.
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Director, Turbopropulsion Laboratory, Code 67Sf Naval Postgraduate School Monterey, California 93940	20
5.	LT J. Ferguson Naval Air Systems Command AIR-536 Washington, D.C. 20361	2
6.	T. H. Okiishi Department of Mechanical Engineering Iowa State University Ames, Iowa 50011	1

199633

Thesis
F25723 Ferguson
c.1

Finite element program for calculating flows in turbomachines with results for NASA Task-1 compressor.